Development of a Novel Algorithm for High Performance Reentry Guidance

Vom Fachbereich Produktionstechnik

der

UNIVERSITÄT BREMEN

zur Erlangung des Grades
Doktor-Ingenieur
genehmigte

Dissertation

von

M. Sc. Marco Sagliano

Gutachter: Prof. Dr. rer. nat. H. Dittus
Fachbereich Produktionstechnik,
Universität Bremen

Dr.-Ing. E. Mooij
Department of Space Engineering,
Delft University of Technology

Tag der mündlichen Prüfung: 22 Jan 2016

Acknowledgements

Since I arrived at DLR in March 2011, there was no day I spent without learning something new. I shared tips, ideas, suggestions and experiences with many of my colleagues, and for this and many other reasons, I'd like to thank them.

A special thanks goes to Dr. Stephan Theil. Everytime we have talked about SHEFEX-3 and not only in those moments, he has always been able to provide smart suggestions, useful insights, and, more in general, a coherent and solid leadership.

I would also want to dedicate a special thanks to Dr. Erwin Mooij, who accepted to supervise my PhD work pro-bono. Since our first meeting, he has given a tremendous and continuous support, together with a genuine interest for the work synthesized in this thesis. His expertise and his suggestions have been a fundamental help during the development of the work.

Many thanks to Prof. Hansjörg Dittus, who has been very supportive. He has dedicated part of his time to this thesis, and has provided insightful suggestions.

Thanks to my greatest cultural references: Fabrizio De André, Pier Paolo Pasolini, Charles Bukowski, and Bill Evans, who have hugely enriched my life.

Thanks to my eminent fellow citizens, Eduardo De Filippo, Roberto Saviano, Peppe Lanzetta, and James Senese, who continuously make me proud of my heritage, and have been a precious source of inspiration in several moments.

Thanks to Naples, which too often forgets his sons, but is never forgotten by them, wherever they are.

A warm thanks goes to my family, especially to my brothers Emilio and Francesco. Many of my loudest laughs occurred and occur when I'm with them.

And finally, the greatest thanks goes to the brightest star in my life, my best friend, my beloved wife Lisa. Without her, I'd have never had the determination to realize all the things I'm proud of, and for this, and many other reasons, I love and I'll love her for the rest of my life.

Non dovrai che restare sul ponte

e guardare le altre navi passare,

le piú piccole dirigile al fiume,

le piú grandi sanno giá dove andare.

Cosí son diventato mio padre,

ucciso in un sogno precedente

il tribunale mi ha dato fiducia,

assoluzione e delitto lo stesso movente.

Fabrizio De André

Abstract

While Europe has done tremendous steps forward in many aspects of space technology, there is still a gap for what regards the development of the technologies for a safe and reliable atmospheric entry. Several US-based programs such as NASA's Orion, SpaceX's Dragon, and Sierra Nevada Corporation's Dream Chaser show that this capability is essential for a fruitful and safe access to space. ESA's successful Intermediate eXperimental Vehicle (IXV) flight shows that Europe can now make important steps in this direction.

In the European frame, the German Aerospace Center plays a main role with the development of the SHEFEX program. While the successful SHEFEX-1 and SHEFEX-2 flights were milestones for experiments associated with navigation and control subsystems, no guidance algorithms for a completely autonomous entry were tested so far. The continuation of SHEFEX, with SHEFEX-3, and its evolution ReFEX, may answer some fundamental questions. Is it possible for a vehicle such as the SHEFEX-3 spacecraft to have a safe and autonomous entry in the presence of a strongly constrained scenario, despite several uncertainties and disturbances acting on the vehicle? Are the standard methods based on drag-tracking (as in the case of the Space Shuttle, or of the aforementioned IXV) still the state of the art, or it is possible to implement new architectures based on the improved capabilities of the modern CPUs? And finally, how can the entry guidance deal with a strong asymmetric scenario? The present work answers these questions by illustrating the strategies conceived for SHEFEX-3, but which can in principle be adopted for future missions, to reduce the mission costs, and at the same time, to increase the reliability and the performance of the atmospheric entry.

In this thesis a dual strategy is proposed. In what we call main guidance, novel methods based on optimal control, adaptive trajectory generation, and sliding-mode control, are developed and tested. In what we call backup guidance, a traditional drag-tracking method is applied to the SHEFEX-3 scenario. Some of the hypotheses usually valid for other scenarios, are here rejected. For this reason, the method developed for SHEFEX-3 can be seen as a generalization of the traditional methods used for entry guidance of vehicles like the Space Shuttle or the IXV. The two guidance methods are completely independent, and can run in parallel. Significant uncertainties

are considered for testing them, and extensive Monte-Carlo campaigns have been run for the verification of the algorithms.

Finally, a proper comparison of the methods is done at the end of the work, together with the conclusions and the lesson learned for the future. Results show that even if both the proposed methods provide good results despite several uncertainties and constraints acting on the vehicle, the main guidance performs better than the backup guidance in terms of final dispersion and range-to-go. For instance, when the perturbed US76 atmospheric model is used, the main guidance allows to have more than 60% of the cases falling within a radius of 50 km w.r.t. the target nominal point, while this percentage decreases to about 55% in case the backup guidance is used. Moreover, when more complex models for the atmosphere are used (e.g., the NRLMSISE-00), 1.6% of the cases associated with the backup guidance exceeds the maximum dynamic pressure, while the main guidance always allows to satisfy the constraints, and therefore, does not threaten the success of the mission. The main guidance proposed here can be therefore considered a valid choice in the frame of the future roadmap for the development of key GNC technologies for atmospheric entry, in addition to the existing methods.

Zusammenfassung

Während Europa (in vielen Aspekten der Weltraumtechnologie) enorme Schritte nach vorne getan hat, so gibt es doch immer noch eine Lücke bei der Entwicklung von Technologien für einen sicheren und zuverlässigen Wiedereintritt. Mehrere US-basierte Programme wie NASA Orion, SpaceX Dragon und Sierra Nevada Corporation Dream Chaser zeigen, dass diese Fähigkeit wichtig ist für einen effizienten und sicheren Zugang zum Weltraum. Der erfolgreiche Flug des "Intermediate Experimental Vehicle" (IXV) der ESA zeigt, dass Europa jetzt wichtige Schritte in diese Richtung machen kann.

Im europäischen Rahmen spielt das Deutsche Zentrum für Luft- Raumfahrt eine Hauptrolle bei der Entwicklung des SHEFEX-Programms (engl. SHarp Edge Flight EXperiment). Während die erfolgreichen SHEFEX-1 und SHEFEX-2 Flüge Meilensteine waren für Experimente mit Navigation und Steuerung wurden bisher keine Flugplanungsalgorithmen für einen autonomen Wiedereintritt getestet. Die Fortsetzung von SHEFEX mit SHEFEX-3 und seiner Weiterentwicklung ReFEX (engl. Reentry Flight Experiment) werden einige grundlegende Fragen beantworten: Ist es möglich für einen Wiedereintrittskrper wie SHEFEX-3 einen sicheren und autonomen Wiedereintritt zu gewährleisten, trotz eines eingeschränkten Missionsszenarios, verschiedener Unsicherheitsfaktoren und Störungen die auf den Krper wirken? Sind die Standardverfahren basierend auf Drag-Tracking (wie im Fall des Space Shuttle oder des vorgenannten IXV) noch "State of the Art", oder ermöglichen leistungsfähigere Onboard-Computer die Verwendung von neuen Algorithmen? Und schließlich: Kann die Flugführung mit einem stark asymmetrischen Szenario umgehen? Die vorliegende Arbeit beantwortet diese Fragen durch die Präsentation der Strategien die für SHEFEX-3 entwickelt wurden, die aber prinzipiell auf zukünftige Missionen angepasst werden können, um die Missionskosten zu reduzieren und gleichzeitig die Verlässlichkeit und Leistungsfähigkeit des atmosphärischen Wiedereintritts zu verbessern.

In dieser Arbeit wird eine zweigleisige Strategie vorgeschlagen. In der sogenannten Haupt-Flugfhrung werden neue Methoden basierend auf optimaler Steuerung, adaptiver Trajektorien Generierung und Sliding-Mode Regelung entwickelt und getestet. In der Ersatz-Flugfhrung wird eine traditionelle Drag-Tracking Methode auf das SHEFEX-3 Szenario angewandt. Einige

der normalerweise geltenden Voraussetzungen sind bei SHEFEX-3 nicht erfüllt, deshalb können die entwickelten Methoden als eine Generalisierung der traditionellen Methoden gesehen werden, die beim Space Shuttle oder IXV benutzt wurden. Beide Flugplanungsalgorithmen sind voneinander unabhängig und können parallel ausgeführt werden. Signifikante Unsicherheiten wurden angenommen, um die Algorithmen zu testen und ausführliche Monte Carlo Simulationen wurden zu ihrer Verifizierung durchgeführt.

Schließlich werden die Methoden am Ende der Arbeit verglichen und entsprechende Schlussfolgerungen über gezogen. Die Ergebnisse zeigen, dass beide vorgeschlagenen Methoden gute Ergebnisse erzielen, trotz der Unsicherheiten und Einschränkung des Fahrzeugs und des Szenarios. Die Haupt-Flugführung bertrifft dabei die Ersatz-Flugführung in Bezug auf die laterale Verteilung und die range-to-go. Zum Beispiel, wenn das gestörte US76 Atmosphärenmodell verwendet wird, kann die Haupt-Flugführung in mehr als 60% der Fälle den Zielpunkt in einem Umkreis von 50 km erreichen, während dieser Anteil auf etwa 55% sinkt, wenn die Ersatz-Flugfhrung verwendet wird. Außerdem, wenn komplexere Modelle für die Atmosphäre verwendet werden (z.B. der NRLMSISE-00) und die Ersatz-Flugführung benutzt wird, wird der maximale dynamische Druck in 1.6% der Fälle überschritten, während die Haupt-Flugführung die Beschrnkungen immer einhält, und somit nicht den Erfolg der Mission gefhrdet. Die vorgeschlagene Haupt-Flugfhrung ist damit eine mögliche Wahl für die zukünftige Weiterentwicklung von GNC-Schlüsseltechnologien für den atmosphärischen Wiedereintritt, neben den bereits existierenden Verfahren.

Contents

Nomenclature

Acronyms

The following acronyms are used:

AA	:	Airspeed-based Aerodynamic reference frame
AB	:	Asymmetric Body
AG	:	Groundspeed-based Aerodynamic reference frame
AMPI	:	Adaptive Multivariate Pseudospectral Interpolation
B	:	Body reference frame
BG	:	Backup Guidance
Cont	:	Continuous
CoM	:	Center of Mass
CL	:	Closed Loop
C.N.	:	Condition Number
CPU	:	Central Processing Unit
DCA	:	Downrange-Crossrange-Altitude
DLR	:	Deutsches Zentrum für Luft- und Raumfahrt
ECEF	:	Earth-Centered Earth-Fixed
EIP	:	Entry Interface Point
FB	:	Feedback
FF	:	Feedforward
FRP	:	Flipped Radau Pseudospectral
GNC	:	Guidance, Navigation, and Control
HD	:	High-Density
HG	:	Hang Glider
HOR	:	Horizontal
HWM07	:	Horizontal Wind Model 2007
IP	:	Inverse-Power
IPOPT	:	Interior-Point OPTimizer
IS	:	Isoscaling
JAC	:	Jacobian
IXV	:	Intermediate eXperiment Vehicle
JRN	:	Jacobian Rows Normalization
LD	:	Low-Density

LOG	:	Logarithmic
LOS	:	Line of Sight
LQR	:	Linear Quadratic Regulator
LVLH	:	Local-Vertical Local-Horizontal
MC	:	Monte Carlo
MG	:	Main Guidance
MIMO	:	Multi-Input Multi-Output
MORABA	:	MObile RAketenBAsis
MPI	:	Multivariate Pseudospectral Interpolation
MRAG	:	Model Adaptive Reference Guidance
MZU	:	Meridional-Zonal-Up
NASA	:	National Aeronautics and Space Administration
NED	:	North-East-Down
NLP	:	NonLinear Programming
NRLMSISE-00	:	Naval Research Laboratory Mass Spectrometer and Incoherent Scatter Radar (from ground to Exosphere) 2000
Num	:	Numerical
OCP	:	Optimal-Control Problem
OL	:	Open Loop
PID	:	Proportional-Integral-Derivative
PJRN	:	Projected-Jacobian Rows Normalization
Ps	:	Pseudospectral
ReFEX	:	Reusability Flying EXperiment
Req-GXYZ	:	Requirement for the Guidance no. XYZ
SHEFEX	:	Sharp-Edge Flying EXperiment
SMC	:	Sliding Mode Control
SNOPT	:	Sparse Nonlinear OPTimizer
SPARTAN	:	SHEFEX-3 Pseudospectral Algorithm for Reentry Trajectory ANalysis
SS	:	Space Shuttle
TA	:	Airspeed-based Trajectory reference frame
TAEM	:	Terminal Area for Energy Management
TG	:	Groundspeed-based Trajectory reference frame
Th	:	Theoretical
US76	:	US Standard Atmosphere 1976
VSC	:	Variable Structure Control
V&V	:	Verification and Validation
WGS84	:	World Geodetic System 1984

Notation

Generally, a lower case variable u is a scalar, a boldface variable $\mathbf{u}$ is a vector, a boldface upper case variable $\mathbf{C}$ is a matrix; a rotation matrix from frame A to frame B is indicated as $\mathbf{T}_A^B$, while a rotation matrix of an angle α around the axis i is indicated as $\mathbf{T}_i(\alpha)$. In case some symbols are doubled, their proper definition is repeated in the right context for better clarity. Additionally notation includes (units specified after the coma, nondimensional if not specified, or with units dependent from the context if indicated with *):

$()_0$	:	Initial value, units of ()
$()_{aa}$	:	Variable expressed in airspeed-based trajectory reference frame*
$()_{ag}$	:	Variable expressed in groundspeed-based trajectory reference frame*
$()_b$	:	Variable expressed in body reference frame*
$()_c$	:	Closed-loop value ()
$()_{const}$	:	Constant value*
$()_e$	:	Variable expressed in ECEF reference frame*
$()_F$	:	Final value, units of ()
$()_{gca}$	:	Great-circle approximation*
$()_{lvlh}$	:	Variable expressed in LVLH reference frame*
$()_L$	:	Lower value, units of ()
$()_{mzu}$	:	Variable expressed in MZU reference frame*
$()_{ned}$	:	Variable expressed in NED reference frame*
$()_o$	:	Open-loop value ()
$()_{ref}$	:	Reference value, units of ()
$()_{ta}$	:	Variable expressed in airspeed-based trajectory reference frame*
$()_{tg}$	:	Variable expressed in groundspeed-based trajectory reference frame*
$()_U$	:	Upper value, units of ()
$\dot{()}$	:	Derivative w.r.t. time, units of ()/s
$\ddot{()}$	:	2$^{\text{nd}}$ derivative w.r.t. time, units of ()/s^2
$\dddot{()}$	:	3$^{\text{rd}}$ derivative w.r.t. time, units of ()/s^3
$()'$	:	Derivative w.r.t. specific energy - units of ()s^2/m^2
$()''$	:	2$^{\text{nd}}$ derivative w.r.t. specific energy - units of ()s^4/m^4
$()^-$	:	Variable generating negative CoM, rad
$()^+$	:	Variable generating positive CoM, rad
$\hat{()}$	:	Side of spherical triangle, rad
$\circ$	:	Hadamard product

Symbols

The following symbols are used throughout the document unless specified differently:

$\mathbf{0_{n\times m}}$	:	$n \times m$ Zero-matrix*
$\mathbf{A}$	:	Dynamic matrix*
$\mathbf{A}$	:	SMC vector not dependent from the control, m/s^3
a_D	:	2$^{\text{nd}}$ derivative of drag w.r.t. energy not dependent from control, m^4/s^4
a_h	:	Component of 3$^{\text{rd}}$ derivative of altitude not dependent from control, m/s^3
a_σ	:	Bank-angle acceleration, rad/s^2
a_V	:	Component of 2$^{\text{nd}}$ derivative of velocity not dependent from control, m/s^3
$\mathbf{B}$	:	Control matrix*
$\mathbf{B}$	:	SMC matrix dependent from the control, m/s^2
$B_i^{j,k}$	:	Experimental spectral coefficient, m/s
b_D	:	2$^{\text{nd}}$ derivative of drag w.r.t. energy dependent from control, m^4/s^4
$b_{h,\alpha}$	:	Component of 3$^{\text{rd}}$ derivative of altitude dependent from α, m/s^2
$b_{h,\sigma}$	:	Component of 3$^{\text{rd}}$ derivative of altitude dependent from σ, m/s^2
$b_{V,\alpha}$	:	Component of 2$^{\text{nd}}$ derivative of velocity modulus dependent from α, m/s^2
$b_{V,\sigma}$	:	Component of 2$^{\text{nd}}$ derivative of velocity modulus dependent from σ, m/s^2
$\mathbf{C}$	:	Output matrix*
$\mathbf{C}$	:	Generalized NLP constraint matrix*
C_D	:	Drag coefficient
$C_{D,a}$	:	Airspeed-based drag coefficient
$C_{D,\alpha}$	:	Partial derivative of C_D w.r.t. α
$C_{D,\alpha\alpha}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. α
$C_{D,\alpha h}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. α and h, 1/m
$C_{D,\alpha M}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. α and M
$C_{D,h}$	:	Partial derivative of C_D w.r.t. h, 1/m
$C_{D,h\alpha}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. h and α, 1/m
$C_{D,hh}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. h, 1/m^2
$C_{D,hM}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. h and M, 1/m
$C_{D,M}$	:	Partial derivative of C_D w.r.t. M
$C_{D,M\alpha}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. M and α
$C_{D,Mh}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. M and h, 1/m
$C_{D,MM}$	:	2$^{\text{nd}}$ partial derivative of C_D w.r.t. M
$C_{D,trim}$	:	Drag coefficient generating $CoM = 0$, rad
C_f	:	Friction coefficient
C_L	:	Lift coefficient
$C_{L,a}$	:	Airspeed-based lift coefficient
$C_{L,\alpha}$	:	Partial derivative of C_L w.r.t. α
$C_{L,h}$	:	Partial derivative of C_L w.r.t. h, 1/m
$C_{L,M}$	:	Partial derivative of C_L w.r.t. M
$C_{L,trim}$	:	Lift coefficient generating $CoM = 0$, rad
$C_{M,O}$	:	Pitch moment coefficient w.r.t. the vehicle's nose

NOMENCLATURE

$C_{M,CoM}$	:	Pitch moment around CoM
$C_i^{j,k}$	:	Experimental spectral coefficient, m/s
C_X	:	Total aerodynamic force coefficient w.r.t. the X axis
$C_{X,B}$	:	Aerodynamic force coefficient of the body w.r.t. the X axis
$C_{X,F}$	:	Aerodynamic force coefficient of the flap w.r.t. the X axis
C_Z	:	Total aerodynamic force coefficient w.r.t. the Z axis
$C_{Z,B}$	:	Aerodynamic force coefficient of the body w.r.t. the Z axis
$C_{Z,F}$	:	Aerodynamic force coefficient of the flap w.r.t. the Z axis
c_{q_c}	:	Constant value of ρV^n, kg/m^{3n}/s^n
CoM_x	:	Center of mass along x axis
CoM_z	:	Center of mass along z axis
$\mathbf{D}$	:	Discrete differentiation matrix, 1/s
$\mathbf{D}_{i,j}$	:	(i^{th},-k^{th}) differentiation submatrix, 1/s
$D_{i,j}$	:	(i^{th},-k^{th}) differentiation element, 1/s
D	:	Drag acceleration, m/s^2
$\bar{D}$	:	Drag force, kg m/s^2
D_a	:	Airspeed-based drag acceleration, m/s^2
D_E	:	Drag-energy constant, m^2/s^2
D_{Eq_g}	:	Drag acceleration corresponding to the equilibrium-glide condition, m/s^2
D_i	:	Drag constant, m/s^2
$D_{n_z,max}$	:	Drag acceleration corresponding to max vertical load factor, m/s^2
$D_{\dot{Q},max}$	:	Drag acceleration corresponding to max heat flux, m/s^2
$D_{\bar{q},max}$	:	Drag acceleration corresponding to max dynamic pressure, m/s^2
d	:	Arc defined by two distinct points over a sphere, rad
d	:	Integer index
E	:	Specific energy, m^2/s^2
E_i	:	Energy constant, m^2/s^2
e	:	Nondimensional energy
e_h	:	Altitude error, m
e_V	:	Velocity modulus error, m/s
$\mathbf{e_y}$	:	Output error*
$\mathbf{F}$	:	Discretized differential equations*
$F10.7$	:	Daily value of measured 10.7 cm solar radio emission, J Hz /m^2/s
$\mathbf{f}$	:	Generic vector function*
$f()$	:	scalar function*
$\mathbf{f}$	:	Nonlinear dynamics function*
f_i	:	i^{th} gain
$\mathbf{G}$	:	Discretized constraints*
$\mathbf{g}$	:	Generic constraint*
$\mathbf{g}_k$	:	k$^{\text{th}}$ nonlinear control-dependent function*
g	:	Gravity acceleration, m/s^2
g_0	:	Gravity acceleration at sea level, m/s^2
$\mathbf{h}$	:	Nonlinear output function*
h	:	Altitude, m

h^*	:	Pseudoaltitude, m
h_s	:	Scale height, m
$\mathbf{I}_k$	:	$k \times k$ unitary matrix
Im	:	Imaginary part*
i	:	Imaginary unit
i	:	Integer index
$\mathbf{J}$	:	Discretized cost function*
$\mathbf{Jac}$	:	Jacobian matrix*
$\mathbf{Jac}_{Num}$	:	Numerical Jacobian matrix*
$\mathbf{Jac}_{PS}$	:	Pseudospectral Jacobian matrix*
$\mathbf{Jac}_{Th}$	:	Theoretical Jacobian matrix*
J	:	Cost function*
J_2	:	2^{nd} zonal coefficient, equal to $1.0826271 \cdot 10^{-3}$
j	:	Integer index
$\mathbf{K}$	:	Gain matrix*
$\mathbf{K}$	:	SMC vector of nonlinear gains, m/s^3
$\mathbf{K}_e$	:	Adaptive gain matrix*
$\mathbf{K_f}$	:	Constraint scaling matrix*
K_γ	:	Pseudoaltitude constant, m
$\mathbf{K}_r$	:	Adaptive Gain Matrix*
$\mathbf{K}_x$	:	Adaptive Gain Matrix*
$\mathbf{K_x}$	:	State scaling matrix*
$\mathbf{K}_u$	:	Adaptive Gain Matrix*
K	:	ballistic coefficient, kg/m^2
K_h	:	SMC nonlinear gain for the altitude, m/s^3
K_V	:	SMC nonlinear gain for the velocity modulus, m/s^3
k	:	Integer index
k_α	:	Feedback gain for the angle of attack, s/m^2
k_D^i	:	Integral gain for 2^{nd}-order drag-energy error dynamics, s^6/m^6
k_h	:	Nonlinear sliding-mode gain for the altitude, m/s^3
k_i	:	i^{th} gain*
$k_{\dot{Q}}$	:	Heat-flux constant, equal to $1.2444 \cdot 10^{-3}$ $\text{kg}^{1/2}/\text{m}^3$
k_V	:	Nonlinear sliding-mode gain for the velocity modulus, m/s^2
$\mathbf{L}_f^i()$	:	i^{th} Lie derivative of () w.r.t. $\mathbf{f}$*
L	:	Lift acceleration, m/s^2
$\bar{L}$	:	Lift force, kg m/s^2
L_a	:	Airspeed-based lift acceleration, m/s^2
L_N	:	Legendre polynomial of order N
M	:	Mach number
$\mathbf{M}_{cons_u}$	:	Dependency matrix of u in the constraint functions
$\mathbf{M}_{cons_x}$	:	Dependency matrix of x in the constraint functions
$\mathbf{M}_{cost_{\Phi,t}}$	:	Dependency matrix of t in the Mayer term
$\mathbf{M}_{cost_{\Phi,u}}$	:	Dependency matrix of u in the Mayer term

NOMENCLATURE

$\mathbf{M}_{cost_{\Phi,x}}$	:	Dependency matrix of x in the Mayer term
$\mathbf{M}_{cost_{\Psi,t}}$	:	Dependency matrix of t in the Lagrange term
$\mathbf{M}_{cost_{\Psi,u}}$	:	Dependency matrix of u in the Lagrange term
$\mathbf{M}_{cost_{\Psi,x}}$	:	Dependency matrix of x in the Lagrange term
$\mathbf{M}_{dyn_u}$	:	Dependency matrix of u in the dynamics functions
$\mathbf{M}_{dyn_x}$	:	Dependency matrix of x in the dynamics functions
$\mathbf{M_{GD}}$	:	NLP dependency matrix
M_h	:	Partial derivative of M w.r.t. h, 1/m
M_hh	:	2^{nd} partial derivative of M w.r.t. h, $1/\text{m}^2$
M_hV	:	2^{nd} partial derivative of M w.r.t. h and V, s/m^2
M_i	:	Molecular weight of i^{th} specie
M_V	:	Partial derivative of M w.r.t. V, s/m
M_Vh	:	2^{nd} partial derivative of M w.r.t. V and h, s/m^2
M_VV	:	2^{nd} partial derivative of M w.r.t. V, s^2/m^2
m	:	Vehicle mass, kg
m	:	Integer index
N	:	Normal distribution
N_A	:	Avogadro number, equal to $6.022169\cdot 10^{26}$ /kmol
$\tilde{N}_{HD}$	:	High-density number of nodes
N_{LD}	:	Low-density number of nodes
n	:	Number of collocation points
n_c	:	Number of continuous controls
n_g	:	Number of continuous constraints
n_i	:	Number density of atoms of i^{th} specie
n_s	:	Number of continuous states
n_z	:	Vertical load factor
$\mathcal{P}$	:	Discrete parameter space*
$\mathcal{P}_c$	:	Continuous parameter space*
$\mathbf{P_{FRP}}$	:	Low-density to High-density conversion matrix
P_i^u	:	i^{th} Lagrange polynomial for the control
P_i^x	:	i^{th} Lagrange polynomial for the state
$\mathbf{p}_i$	:	i^{th} set of discrete parameters*
p	:	Atmospheric pressure, N/m^2
$\mathbf{Q}$	:	Output-error weighting matrix*
$\dot{Q}$	:	Heat flux, W/m^2
$\bar{q}$	:	dynamic pressure, N/m^2
$\mathbf{R}$	:	Control-effort weighting matrix*
$\mathbb{R}^d$	:	d-dimensional set of real numbers
$\mathbb{R}_i$	:	subset of real numbers
R	:	Range, m
R^*	:	Universal gas constant, equal to $8.31432\cdot 10^{-3}$ N m/ kmol K
$R_\oplus$	:	Earth's gravitational parameter, equal to 6378.100 km^3/s^2
R_N	:	Radau polynomial of order N

$\mathbf{r}$	:	Augmented error*
$\mathbf{r}$	:	Relative degree of the system
r	:	Radial position, m
Re	:	Real part*
S	:	Vehicle surface, m^2
s_h	:	Altitude sliding variable
s_{lat}	:	sign of the bank angle
s_V	:	Velocity modulus sliding variable
sat	:	Saturation function
sign	:	Sign function
T	:	Temperature, K
$\tilde{\mathbf{T}}_{HD}$	:	High-density discrete trajectory*
$T_{h,i}$	:	i^{th} temperature gradient, K/m
T_i	:	Constant i^{th} temperature, K
$\mathbf{T}_{LD}$	:	Low-density discrete trajectory*
t	:	time, s
$\mathbf{U}_{LD}$	:	Low-density discrete controls*
$\tilde{\mathbf{U}}_{HD}$	:	High-density discrete controls*
$\mathbf{U}_i$	:	i^{th} discrete control*
$\mathbf{U}_m$	:	Mean value of discrete controls*
$\mathbf{U}_p$	:	Perturbed value of discrete controls*
$\mathbf{u}$	:	Generic control*
$\mathbf{u}$	:	Vector SMC control, m/s
$\mathbf{u}_i$	:	i^{th} wind component, m/s
$\mathbf{u_m}$	:	Model input*
$\mathbf{u_p}$	:	Plant input*
u_α	:	Angle-of-attack rate, rad/s
u_σ	:	Bank-angle rate, rad/s
$u_{\sigma,max}$	:	Maximum value of bank-angle rate, rad/s
u_{FB}	:	Feedback control
u_{FF}	:	Feedforward control
$\mathbf{V}_a^{ned}$	:	Airspeed-based velocity vector in NED reference frame, m/s
$\mathbf{V}_g^{ned}$	:	Groundspeed-based velocity vector in NED reference frame, m/s
$\mathbf{V}_w^{mzu}$	:	Wind-velocity vector in MZU reference frame, m/s
$\mathbf{V}_w^{ned}$	:	Wind-velocity vector in NED reference frame, m/s
V	:	Velocity modulus, m/s
V_a	:	Airspeed-based wind velocity modulus, m/s
V_g	:	Groundspeed-based wind velocity modulus, m/s
V_h	:	Lyapunov function for the altitude
V_V	:	Lyapunov function for the velocity modulus
V_w	:	Wind velocity modulus, m/s
$\tilde{\mathbf{X}}_{HD}$	:	High-density discrete states*
$\mathbf{X}_i$	:	i^{th} discrete state*

$\mathbf{X}_{LD}$	:	Low-density discrete states*
$\mathbf{X}_m$	:	Mean value of discrete states*
$\mathbf{X}_{NLP}$	:	NLP state vector*
$\mathbf{X}_p$	:	Perturbed value of discrete states*
$\mathbf{x}$	:	Generic state*
$\mathbf{x_m}$	:	Model error*
$\mathbf{y}$	:	Generic output*

Symbols - Greek

Finally, the following greek symbols have been used:

α	:	Angle of attack
α_a	:	Airspeed-based angle of attack, rad
α_g	:	Groundspeed-based angle of attack, rad
β	:	Sideslip angle, rad
β_a	:	Airspeed-based sideslip angle, rad
β_g	:	Groundspeed-based sideslip angle, rad
$\nabla()$	:	Difference of () w.r.t. reference value*
$\Delta\psi$	:	Heading error, rad
$\Delta\psi_{limit}$	:	Maximum allowed heading error, rad
$\delta()$	:	Variation w.r.t. the nominal value
$\delta_{m,n}$	:	Kronecker delta
δ_{width}	:	Width of the layer for the saturation function*
γ	:	Flight-path angle, rad
γ_{air}	:	Specific heat ratio of the air, assumed to be equal to 1.4
γ_a	:	Airspeed-based flight-path angle, rad
γ_g	:	Groundspeed-based flight-path angle, rad
γ_w	:	Wind flight-path angle, rad
$\mathbf{\Lambda}$	:	SMC linear gain matrix, m/s^3
Λ_h	:	SMC linear gain associated with altitude, m/s^3
Λ_V	:	SMC linear gain associated with velocity modulus, m/s^3
λ_i^h	:	i^{th} Linear sliding-mode gain for the altitude*
λ_i^V	:	i^{th} Linear sliding-mode gain for the velocity modulus*
$\mu_\oplus$	:	Earth's gravitational parameter, equal to 3986004.418 10^8 km^3/s^2
$\omega_{()}$	:	Weight for () in the cost function
$\omega_\oplus$	:	Earths angular rate equal to 7.292115 10^{-5} rad/s
ω_D	:	Natural pseudofrequency of 2^{nd}-order drag-energy error dynamics, s^2/m^2
$\tilde{\omega}_i$	:	i^{th} Radau quadrature weight
ω_σ	:	Bank-angle acceleration, rad/s^2
Φ	:	Lagrange term of cost function*
ϕ	:	Geocentric latitude, rad
Ψ	:	Mayer term of cost function*

$\Psi_i^{j,k,l}$	:	Fourier basis function
ψ	:	Velocity-azimuth angle, rad
ψ_a	:	Airspeed-based velocity-azimuth angle, rad
ψ_{HOR}	:	Angle between the direction of motion and the local horizontal, rad
ψ_g	:	Groundspeed-based velocity-azimuth angle, rad
ψ_{LOS}	:	Angle between the direction of motion and the target line of sight, rad
ψ_w	:	Wind velocity-azimuth angle, rad
ρ	:	Atmospheric density, kg/m^3
σ	:	Bank angle, rad
σ_a	:	Airspeed-based bank angle, rad
σ_g	:	Groundspeed-based bank angle, rad
σ_{cur}	:	Current value of bank angle, rad
σ_u	:	Standard deviation for the control u, units of u
σ_x	:	Standard deviation for the state x, units of x
τ	:	Pseudospectral time
τ	:	Flap deflection, rad
$\tilde{\tau}_i$	:	High-density i^{th} discrete pseudospectral time
τ_i	:	Low-density i^{th} discrete pseudospectral time
τ_{trim}	:	Flap deflection generating $CoM = 0$, rad
θ	:	Geocentric longitude, rad
ζ_D	:	Damping ratio of 2$^{\text{nd}}$-order drag-energy error dynamics

Chapter 1

Introduction

1.1 Scope

Atmospheric Reentry has always been a hard problem for engineers since the beginning of the first space missions. The challenge is made hard by all the constraints that must be taken into account, such as dynamic pressure, load factor or heat flux. Other limits come from the control of the vehicle that, in absence of propulsion, is entirely provided by the aerodynamic forces generated by the vehicle's surfaces. These difficulties are made harder by further restrictions, like the impossibility to vary the bank angle over the entire range $[-180, +180]$ deg, or restrictions on the modulation of the angle of attack. Therefore, we have to deal with an intrinsically control-bounded system.

Initial errors, disturbances, and uncertainties will affect the results, and hence, must be taken into account. Moreover, the environment brings further complexity. The environment, which characterizes the mission scenario (gravity, atmosphere, etc.) is made by mathematical models. They are functional representations of physical variables deduced on the basis of observations and theoretical fundamentals. This means that they can significantly vary during the real mission. Therefore, during the real flight some of the hypotheses may not be fully satisfied, and other variations not taken into account in the models (daily variations, weather conditions, solar activity, and so on) may play an important role. Other effects, like non-linearities or non-modeled phenomena could arise, generating different effects with respect to what expected. Another issue is related to the coefficients of lift and drag, that for experimental flights are hard to quantify with negligible margin of error. Realistic margins need to be included in the verification process, together with other uncertainties, like the mass of the vehicle at the entry interface. Finally, the wind can affect the results as well.

Over the last decades, several methods have been developed, and successfully employed. One of the main branches is related to the drag-velocity (further evolved in the modern drag-energy tracking methods). These methods were originally developed by Bogner[1] for the Apollo entry guidance, and found their greatest success in the many missions accomplished by the Space Shuttle[2], and in the latest NASA successes, such as the Mars Exploration Rovers, and the Mars Science Laboratory[3,4] programs.

The principle behind this family of methods resides in the fact that all the classical constraints previously cited, such as the dynamic pressure, the heat flux, and the load factor, together with the range, can be efficiently represented in the drag-velocity, and later, in the drag-energy plane. In this plane, a reference drag-profile, taking these constraints into account, can be designed and tracked with an ad-hoc tracking controller. The longitudinal dynamics of the vehicle is in this way completely determined. In addition, the heading angle (and as a consequence, the lateral error) is kept under control with a proper management of the bank-angle sign.

However, given the increased capabilities of the CPU, a second branch of methods, based on optimal-control theory, has arisen. In this case the differential equations describing the problem, together with the constraints, can be properly discretized and a guidance solution can be generated according to the maximization (or minimization) of a given cost index. Even though these technologies are becoming more and more mature, their use is still limited by the fact that, for some problems, it is not possible to determine the number of operations required to compute a solution. This makes the adaptation of the solution to off-nominal conditions, disturbances, or uncertainties difficult. This is the case of the entry guidance problem. In order to make a step ahead in the integration of the optimal control theory with the adaptive and robustness capabilities required for such a complex problem, the entry guidance method developed in this work has been proposed. The main question of this study has been formulated as follows:

How can optimal control theory be integrated into an entry guidance scheme, which is real-time capable, and able to deal with a highly-constrained scenario? If successful, is this method able to compete with the current state of the art for the entry guidance algorithms (i.e., the drag-tracking methods)?

The scenario considered here is the one proposed for SHEFEX-3, where SHEFEX stands for Sharp Edge Flying EXperiment, a technological program led and developed by DLR. The asymmetry of the mission, coming from the requirement to show high crossrange capability, together with the constraints of the scenario and the vehicle, has made the entry guidance problem a tough, and at the same time, stimulating challenge. To measure the performance of the developed guidance scheme, a dual strategy has been pursued. In what we will call the main guidance scheme, the optimal control theory, based on the use of pseudospectral methods, are merged with adaptive algorithms for the on-board generation of the feedforward solution, and the use of nonlinear techniques for the feedback controller. In the so-called backup guidance, the idea has been to reuse the drag-tracking approach, which has been successfully tested in several mission, in virtue of its reliability, and easiness of implementation. However, modifications have been adopted to take the nature of SHEFEX-3 mission into account. The Monte-Carlo campaigns associated with the single guidance methods will answer the first part of the question previously formulated. At the end of the work, a comparison between the two guidance schemes is performed and discussed, such that the second part of the question

can find a satisfactory and unique answer.

1.2 Summary of Contributions

This work contributes to the field of entry guidance in many ways. First, this work deeply applies a modern optimization technique such as pseudospectral methods to a realistic, highly-constrained mission involving atmospheric entry. Guidance of successful and well-known missions like the Space Shuttle and Mars Science Laboratory, for instance, find their origin in reliable, but non-optimal methods, such as the drag-velocity tracking scheme. New technological developments in the last twenty years allow to move forward and to introduce algorithms for the generation of optimal trajectories in the loop, to increase the performances of current and future space missions. While pseudospectral methods found their first real application in the zero-propellant reorientation of the International Space Station[5], their use has not yet been proposed for missions involving a planetary guided entry. On this purpose, a brand new tool has been developed. SPARTAN (SHEFEX-3 Pseudospectral Algorithm for Reentry Trajectory ANalysis) implements a specific technique belonging to the pseudospectral methods family, the global flipped Radau method, to compute optimal trajectories tailored for SHEFEX-3. Specific new features have been developed for this work, like advanced techniques for numerical differentiation, automatic scaling routines for the optimal control problems, and exploitation of the Jacobian matrix representing the problem to be solved. This tool has been used for the generation of all the optimal trajectories computed during the development of this work. However, its general-purpose structure makes it a useful tool for other projects and scenarios, as already done with the generation of Lunar-landing reachability sets, or with the generation of optimal trajectories for completely different vehicles, like the X-33 or the NASA Space Shuttle. SPARTAN is, in the current knowledge of the author, the first European optimal control tool based on pseudospectral methods. From this perspective, this tool will help to reduce the gap of Europe w.r.t. the United States, where several research teams and companies work on this novel technique[6].

However, while the CPU power is continuously increasing, the real-time solution of non-convex optimal problems (like the entry guidance problem, indeed) is in many cases still outside the capabilities of the current on-board systems. Therefore, the optimization tool is not sufficient if taken as stand-alone. For this reason, the second objective of this work has been the development of an algorithm able to generate on-board suboptimal trajectories by combining the database of optimal trajectories previously computed, and efficiently stored on-board. The exploitation of the pseudospectral methods led to the possibility to combine a classical multivariate interpolation technique with properties belonging to pseudospectral methods. This leads to a new algorithm, able to generate online entry guidance solutions tailored to the real, off-nominal inflight conditions. The result is a method, which will efficiently store a large number of trajectories, and will combine them to provide an adaptive feedforward entry guidance solution.

Furthermore, other disturbances and uncertainties may affect the flight of the

vehicle. For this reason, a nonlinear feedback controller based on sliding mode control theory has been developed. The controller developed is of general application, and may represent a good alternative to the existing methods. However, the work performed in this thesis does not reject the precious heritage coming from the classical drag tracking approach, which has been implemented here as backup-guidance scheme, to provide further redundancy, and, at the same time, to compare the novel method with the state of the art, that is, the drag-tracking method. The backup-guidance method also includes some improvements. The original Space-Shuttle formulation cannot be applied tout-court to SHEFEX-3, because of the type of mission and the models used. Therefore, in the formulation, which has been proposed here many of the simplifications used in most of the literature[7,8,9] are rejected to make the obtained results more accurate.

What has not been done in the frame of this work is the prototyping of the algorithms in C to perform real-time runs, and their implementation for hardware-in-the-loop tests. The integration of the guidance with the attitude control system has not been considered, and the effects of the control on the guidance design are taken into account in terms of limitations on the angle of attack, bank angle, and their rates. This integration of the guidance and control modules may indeed be a fruitful future research topic, and is out of scope of this thesis.

In terms of assumptions, it is important to stress that variations of the uncertainties and the disturbances may affect the performance of the entry-guidance algorithms discussed in this thesis, and new analysis, based on updated constraints, may be necessary. Therefore, all the results, the analysis, and the conclusions made throughout this thesis are valid in the frame of the range considered for dispersions, uncertainties, and disturbances introduced in Chapter 11. Finally, assumptions of failures of one or more subsystems are here completely excluded, except for one of the two guidance systems, which, as stated, can run in parallel, and are completely separated.

The following algorithms were developed specifically for this work and can be used for future research:

- SPARTAN - (SHEFEX-3 Pseudospectral Algorithm for Reentry Trajectory ANalysis): A matlab-based tool used for the generation of the reference optimal trajectories and the feedforward controls.
- AMPI - (Adaptive Multivariate Pseudospectral Interpolation): An algorithm for the real-time computation of suboptimal trajectories and controls for entry scenarios based on the fusion of the multivariate interpolation and pseudospectral methods.
- A robust, nonlinear controller based on sliding mode control for the tracking of the trajectories, able to work in the presence of uncertainties and disturbances.
- An extensive software library including different atmospheric models (e.g., exponential model, and US76), gravity models with different accuracies, smooth and discrete aerodynamics database, and entry flight models valid under different assumptions.

During the last two years the author led the work of several students (one Master's Degree thesis and three internships) to extend the capabilities of SPARTAN. The work was focused on several problems taken from literature, and brought the publication of the following works, supervised and coauthored by the author (full details are here omitted for brevity, and can be found in the bibliography):

- "Exact Hybrid Jacobian Computation for Optimal Trajectory Computation via Dual Number Theory"[10].
- "SPARTAN: An Improved Global Pseudospectral Algorithm for High-fidelity Entry-Descent-Landing Guidance Analysis"[11].

As previously said, SPARTAN was used for other scenarios as well. This work is summarized in the following publications, coauthored by the author:

- "Safe Landing Area Determination for a Moon Lander by Reachability Analysis"[12].
- " Approximation of Attainable Landing Area of a Moon Lander by Reachability Analysis"[13].

Finally, the following papers, related to this work, were published by the author:

- "Hybrid Jacobian Computation for Fast Optimal Trajectories Generation"[14].
- "Performance Analysis of Linear and Nonlinear Techniques for Automatic Scaling of Discretized Control Problems"[15].
- "SHEFEX-3 Optimal Feedback Guidance"[16].
- "Real Time Adaptive FeedForward Guidance for Entry Vehicles"[17].
- "Onboard Trajectory Generation for Entry Vehicle via Adaptive MultiVariate Pseudospectral Interpolation"[18].

Full information about all the publications listed here can be found in the bibliography, at the end of the thesis.

1.3 Outline of the thesis

The dissertation is arranged as follows:

Chapter 2 provides a description of SHEFEX program, with a focus on SHEFEX-3. This includes the mission profile and the vehicle characterization, as well as the mission and guidance requirements.

Chapter 3 reports the mathematical description for the environmental models used for atmosphere, gravity, wind, and the aerodynamics of the vehicle.

Chapter 4 describes the several flight-mechanics models implemented, specifying the differences, and the assumptions made for each of them.

Chapter 5 introduces the background for this work, with the state of the art for what regards the entry guidance problem.

Chapter 6 reports some examples of the verification and validation methods followed for the several models implemented (e.g., atmosphere, aerodynamics).

Chapter 7 discusses the theoretical foundation, the structure, the features, and the validation of SPARTAN, which was used as reference tool for the generation of all the optimal trajectories.

Chapter 8 describes the implementation of the adaptive feedforward guidance, conceived and developed for SHEFEX-3, based on the development of the adaptive multivariate pseudospectral interpolation (AMPI) technique.

Chapter 9 presents the nonlinear feedback controller developed for SHEFEX-3, based on sliding mode control. Stability and applicability of the algorithms are discussed.

Chapter 10 describes the backup guidance, developed on the basis of the drag-tracking control, similar to the Space-Shuttle entry guidance.

Chapter 11 describes the Monte-Carlo campaign simulations performed to verify the methods developed. The integration of the equations of motion in presence of wind profiles is explained, together with the uncertainties taken into account, and a comparison between the performance of main and backup guidance is reported and discussed.

Finally, some conclusions, with a special emphasis on the lesson learned, and recommendations for the future are the subject of the Chapter 12.

Part I

Mission, Models, State of the Art

Chapter 2

SHEFEX-3 - Vehicle, Mission, and Scenario

The aim of this chapter is to describe the SHEFEX-3 mission. SHEFEX is a program of technological development for atmospheric entry led by the Deutsches-Zentrum für Luft- und Raumfahrt. A short historical note about the origin, and the evolution of the SHEFEX program is reported in Sec. 2.1. The several developments over the year brought to the vehicle prototype for SHEFEX-3, which is described in Sec. 2.2, together with its main characteristics. The characterization of the vehicle is important to understand the kind of reentry that can be performed. For this reason, comparisons with other entry vehicles in terms of control capability help to understand their differences, and their way to approach the entry.

Since we deal with a non-powered vehicle, only the modulation of the aerodynamic forces can be used to control the atmospheric entry. We will see that some vehicles only use bank-angle modulations, such as the Mars Science Laboratory. Other vehicles, such as the Space Shuttle, or the X-33, take advantage from the modulation of the bank angle and the angle of attack at the same time. This leads to the possibility to have a larger control capability to modulate the L/D ratio, and therefore, to increase the range of the vehicle. However, this also brings a larger heat load acting on the vehicle. Therefore, some trade-offs are required, and design choices are driven by the requirements of the specific mission we are dealing with. Following this logic, Sec. 2.3 focuses on the mission scenario used for the development of the guidance algorithms, while a description of the guidance system requirements is reported, together with the constraints coming from the related subsystems, i.e., the navigation and the control subsystems, in Sec. 2.4.

2.1 SHEFEX program

SHEFEX is a program of technological development for atmospheric entry conceived and led by the German Aerospace Center over the last 20 years. Indeed, the first ideas, which later would have brought to the genesis of the SHEFEX program, can be

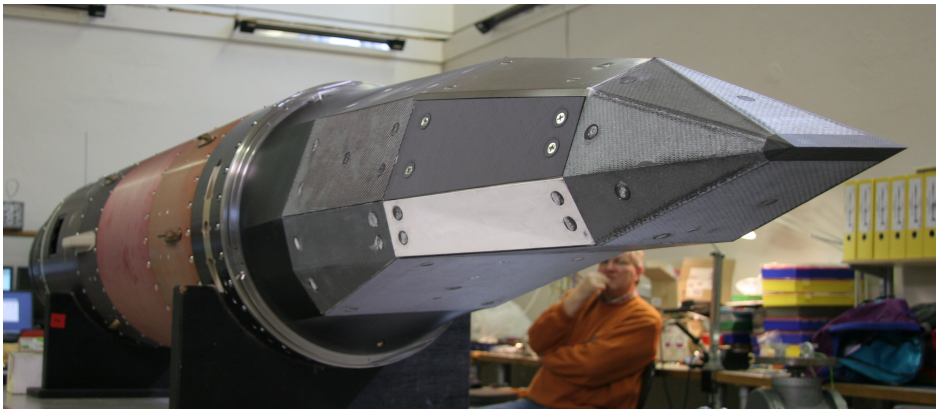


Figure 2.1: SHEFEX payload vehicle[21].

found in the Colibri project, a re-entry demonstrator conceived in 1994[19]. The ideas to test technologies for atmospheric entry such as structure and thermal protection systems were further developed in the Hot Structures Program, from 1994 to 1997. The key-idea to transform blunt areas into flat surfaces, to reduce costs without penalizing performances, led to the first SHarp Edge Flight EXperiment (SHEFEX), successfully launched on October 27, 2005 from Andøya Rocket Range in Andenes, Norway[20]. The main goal of the project was to test the new shape for hypersonic vehicles while the payload vehicle (Fig. 2.1) served as a testbed for new technologies and contained several passenger experiments. SHEFEX-1 acted as a "flying wind tunnel", and used affordable sounding rocket for the launch. Among the technologies to be tested there was the new TPS design, and the testing of sharp leading edge design for entry vehicles. The experiments measured several aerodynamics parameters, and their effect on the structure during atmospheric entry and used passive control during entry. This mission also demonstrated that sounding rockets are suitable for atmospheric entry experiments.

The SHEFEX project served as a starting point for the second SHarp Edge Flight EXperiment (SHEFEX-2), launched on June 22, 2012 from Andøya Rocket Range. The goal of SHEFEX-2 was to validate analytical predictions and ground test data and to investigate technologies for hypersonic and space transportation systems. Again, the shape of the payload vehicle used faceted surfaces and sharp edges, but it was also rotationally symmetric using what was learned from the previous SHEFEX mission (Figure 2.2). Additionally, there was an array of passenger experiments, which investigated thermal-protection-system technologies, active entry-control using canards, antenna technologies, and vehicle navigation. The payload was mounted on a modified Brazilian VS-40 launch vehicle. This solid-propellant sounding rocket used an S-40 and an S-44 for the first and second stage motors. All necessary rocket subsystems were integrated within the payload segments, including: navigation, power, reaction control, communications, parachute and recovery, etc.

Figure 2.2: SHEFEX-2 payload vehicle[21].

The SHEFEX-2 rocket was unguided during the propelled flight phases and actively controlled during entry with the canards. This will change with SHEFEX-3 and its evolution, ReFEX, which, for the first time, will have a completely autonomous guided entry.

2.2 SHEFEX-3 / ReFEX

To go on with the effort to increase the technological level for real space missions, a new challenge in the next years with the development of SHEFEX-3 and ReFEX arises. SHEFEX-3 and ReFEX will increase the technological readiness level of the German Aerospace Center for what regards entry technology. In the next sections a brief look at the vehicle's evolution and its properties will be given.

2.2.1 SHEFEX-3 Vehicle Description

The SHEFEX-3 vehicle has a faceted surface, as its predecessors, which guarantees minor costs in terms of manufacturing. The vehicle has a nominal mass of 500 kg. One of the proposed designs (the reference one for the current work) is shown in Fig. 2.3. The reference surface is equal to 0.468 m^2, and has its center of mass (CoM) at 55%, starting from the nose. Data is reported in Table 2.3.

Figure 2.3: SHEFEX-3 Entry vehicle[19].

Table 2.1: SHEFEX-3 vehicle parameters.

Parameter	Value	Unit
mass	500	kg
surface	0.468	m^2
length	1.85	m
width	1.85	m
height	0.66	m
CoM	55%	-

2.2.2 ReFEX Vehicle Description

Alternative possibilities are being considered, in virtue of the new direction that the ReFEX program is pursuing. An example of the concept is reported in Fig. 2.4. Also in this case, the vehicle will have a mass of 500 kg, and a slightly wider surface, while the shape is biconic. This configuration is at the moment subject of study. However, for the rest of the work, we will refer to the SHEFEX-3 vehicle for the development of the algorithms. They will be then migrated to ReFEX once that the new mission concept has been fully defined, but is not part of the current work.

For what concerns the flight control system, given the nature of the mission and the budget requirements, the vehicle has stricter requirements w.r.t. other entry vehicles, like the X-33 vehicle, or the Mars Science Laboratory, depicted in Figs. 2.5 and 2.6. In particular, the angle of attack and the bank angle have strong limitations w.r.t. the X-33, which is maybe the closest vehicle to SHEFEX-3 as mission profile. Moreover, the angle-of-attack and the bank-angle rate have lower values w.r.t. the cited vehicles. To

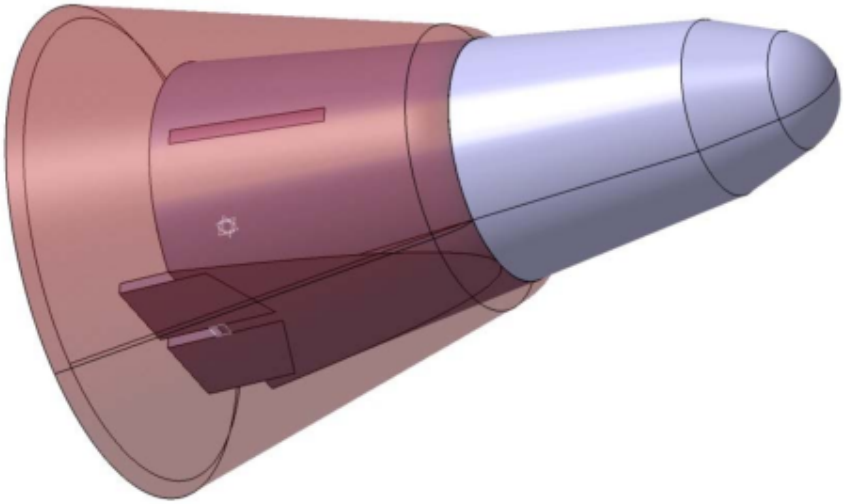


Figure 2.4: ReFEX Entry vehicle.

Figure 2.5: X-33 - Courtesy of NASA and Lockheed Martin.

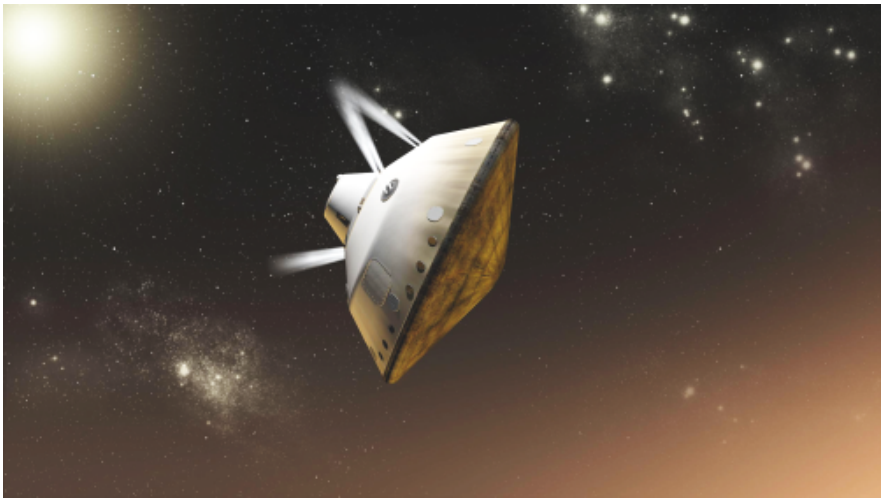


Figure 2.6: MSL - Courtesy of NASA Jet Propulsion Laboratory.

give an idea about these differences, the control authority for the X-33 and SHEFEX-3 are reported in Figs. 2.7 and 2.8. The circles on the left side represent the capability to modulate the absolute bank angle, starting from the condition $\sigma = 0$, corresponding to the local vertical axis. In this condition, the lift force is directed upwards. The circles on the right side represent the capability of the vehicles to change their angle of attack w.r.t. a pre-defined nominal profile.

It is possible to see that while vehicles like the X-33 can vary the bank angle between -108 and 180 deg, SHEFEX-3 will be able to modulate its bank angle in the interval -60 to 60 deg. Further limitations come from the asymmetry of the scenario, as we will see in Chapter 8. Similar considerations can be made for the angle of attack; in both cases there is a reference profile which can be changed. However, this change can be up to $\cong \pm 25$ deg for the X-33, while for SHEFEX-3 the maximum variation allowed is $[-2.5,\ 10]$ deg w.r.t. the reference angle of attack α_{ref}. Also in terms of control rates, strong requirements are to be taken into account. Indeed, both the angle-of-attack rate and the bank-angle rate cannot exceed 5 deg/s, according to the requirements provided by the attitude control system responsible. This is significantly less than what the X-33 can do, since this vehicle has a maximum control rate up to 20 deg/s for both the angle of attack and the bank angle[22].

To realize the on-board experiments, a high, constant angle of attack (equal to 42 deg) is maintained for the first 58 seconds of the entry phase. After this period, a linear transition (which takes about 30 s) from the high angle of attack to a lower angle of attack (17.5 deg) is performed to increase the C_L/C_D ratio, and therefore, to increase the range of the vehicle. Variations w.r.t. the nominal angle of attack are allowed in the limits previously defined. All the flight-control system constraints are listed in Table 2.2.

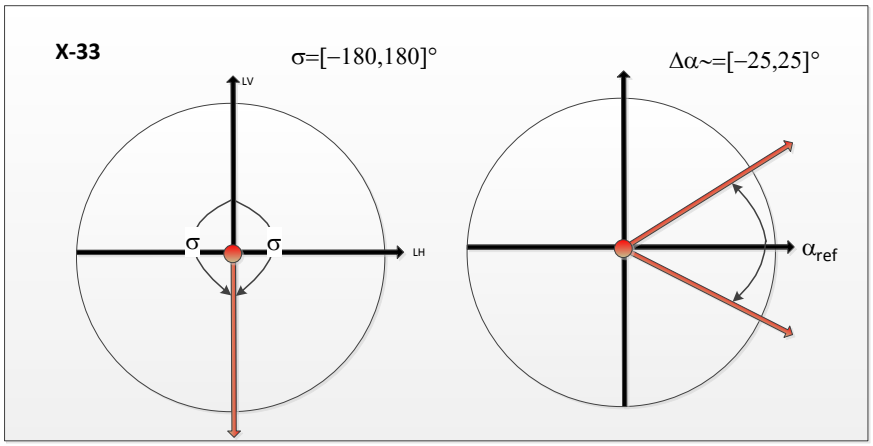


Figure 2.7: X-33 control authority.

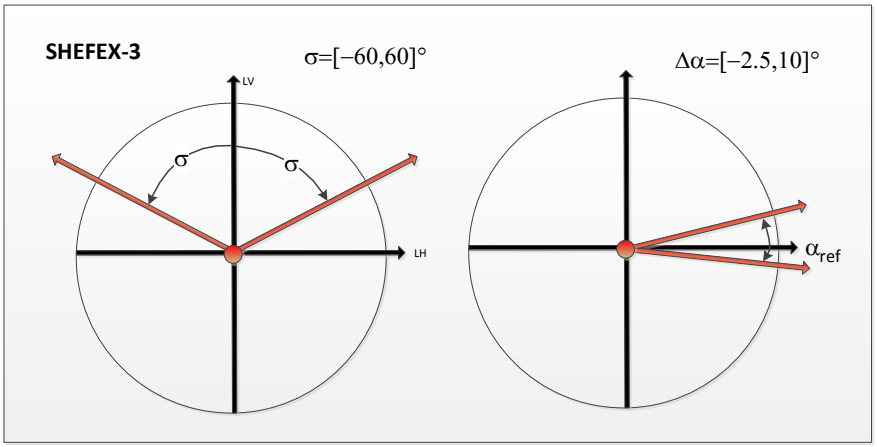


Figure 2.8: SHEFEX-3 control authority.

Table 2.2: Flight Control System Constraints.

Controls	Values / Ranges
Upper angle of attack α_U (deg)	42
Lower angle of attack α_L (deg)	17.5
Begin of α maneuver $t_{\alpha,U}$ (s)	58
End of α maneuver $t_{\alpha,L}$ (s)	88
Bank angle σ (deg)	[-60, 60]
Angle of attack rate $\dot{\alpha}$ (deg/s)	[-5, 5]
Bank angle rate $\dot{\sigma}$ (deg/s)	[-5, 5]

2.3 Mission Scenario

Let us define the specific mission tailored for SHEFEX-3. Also in this case, as in the previous SHEFEX missions, the current launch site is Andøya, while the terminal area is placed in Greenland. The terminal position is depicted in Fig. 2.9.

Figure 2.9: Terminal Area Energy Management - Credits: Google Maps.

An alternative scenario with the terminal point in the Svalbard Archipelago has also been proposed[16]. The spacecraft would be launched with a rocket based on the Brazilian engines S50/S44. After the stages separation and the coast phase, the non-powered descent phase follows. An overview of the SHEFEX-3 mission profile is depicted in Fig. 2.10. Once an altitude of 100 km is reached, the nominal entry phase begins. The entry interface is characterized by a steeper flight-path angle and a lower Mach number w.r.t. other entry missions, like the Space Shuttle reentry or the X-33 studies[7,8,22]. The mission, from the point of view of the guidance system, terminates at the Terminal Area Energy Management (TAEM) interface. The interface is, for this mission, associated with a Mach number equal to $\cong 2$. A strong difficulty comes from the asymmetry of the scenario, chosen to stress the cross-range capabilities of SHEFEX-3.

The requirements in terms of dispersions at the terminal-area interface are associated with a circle having radius equal to 150 km in the worst case, with the center placed in the nominal terminal condition. Moreover, two other circles, with radii equal to 50 and 100 km, are defined to characterize the accuracy of the results. In Table 2.3, the conditions at the Entry interface, and at the TAEM are shown. The objective of the mission, from the point of view of the guidance, is to steer the vehicle towards the TAEM while satisfying all the requirements, defined in the next section, and in presence of uncertainties, disturbances, and initial dispersions.

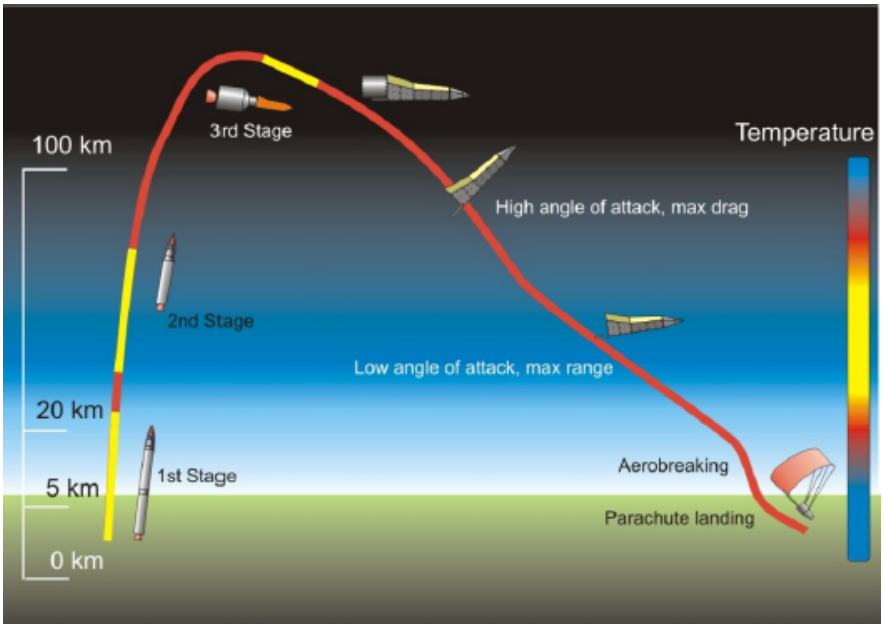


Figure 2.10: SHEFEX-3 Mission Profile[17].

Table 2.3: Nominal Entry and Terminal Conditions for SHEFEX-3 guided flight.

State	Initial Value	Terminal Value
Geocentric Altitude h (km)	100.10	23.5 ± 2.5
Geocentric Longitude θ (deg)	-11.60	-45.75
Geocentric Latitude ϕ (deg)	71.89	66.4
Velocity Modulus V (m/s)	4712.26	595 ± 25
Flight-path Angle γ (deg)	-10.31	free
Velocity Azimuth Angle ψ (deg)	-85.92	free

2.4 Mission Requirements

In this section all the requirements defined for the entry guidance of SHEFEX-3 are reported. In Chapter 12 a proper requirements compliance matrix is reported to ensure that all the requirements are satisfied. These requirements are listed in Table 2.4. This list represents the trade-off among several subsystems and departments involved in the design of the mission. Requirements G001-G003 come from the responsible of structures. The requirements G004-G013 come from the DLR's system analysis and space transportation department, and from the department responsible for the launch and the ascent trajectory. The requirement G014 comes from the application of the ESA policy for the margins regarding the residuals of propellant[23]. Requirement G015 and G018 have been selected after having analysed several models, and the measurements

used as data to build them (e.g., HWM07[24] and NRLMSISE00[25], while the requirements G016 and G017 have been selected on the basis of wind tunnel tests performed by DLR aerodynamics experts[26]. Requirements G019-G026 come from the control subsystem requirements. For the design phase, the US Standard Atmosphere 1976, described in Sec. 3.1.2 model was chosen as trade-off between the exponential model and the NRLMSISE00. The former, which is reported in Sec. 3.1.1, is not accurate enough in the upper region of the atmosphere of our interest, (e.g., in the range [80-100] km), while the latter, which is more computationally expensive, was used for the Monte-Carlo campaign, together with a perturbed version of the US76 model. These reasons justify the requirements G027-G029.

Another trade-off has been the reason for the choice of the central gravity field with the contribution of the J_2 term (Reqs. G030,G031), described in Sec. 3.2.2, as it has been observed that further terms such as J_3 or J_4 do not have any significant impact on the mission, given its time frame (which is in the order of 10 minutes). Given the semistochastic nature, wind gusts coded according to HWM07 model (3.3.1) have not been introduced in the design phase. However, since the no-wind condition in practice does not exist, its influence has been included in all the Monte-Carlo campaigns (Reqs. G032,G033).

In terms of the aerodynamics, the nominal database, computed as explained in Sec. 3.4.6, has been used for the design. However, to test the robustness of the guidance systems, perturbations have been introduced and included in the Monte-Carlo runs. These perturbed profiles have been generated according to the indications provided by several wind tunnel tests[26] performed by DLR aerodynamics specialists. These aspects drove the requirements G034 and G035. Finally, G036 has been selected at the beginning of the work to increase the reliability of the guidance system. The two schemes developed can potentially run in parallel, have no common parts, and are based on completely different concepts and algorithms. As stated, the table below is compared in Chapter 12 with a requirements compliance matrix. The analysis of this matrix will give the possibility to discuss if and how these requirements have been satisfied, and in case some of them have not, the reason why this happened.

Table 2.4: SHEFEX-3 Guidance Requirements.

Id	Description
Req-G001	The dynamic pressure shall not exceed 50 KN/m^2
Req-G002	The heat-flux shall not exceed 6.5 MW/m^2
Req-G003	The vertical load factor shall not exceed 10 g
Req-G004	The Mach number at the TAEM shall be equal to 2
Req-G005	The altitude error at the TAEM shall be within ±2.5 km w.r.t. a nominal value of 23.5 km
Req-G006	The velocity error at the TAEM shall be within ±25 m/s w.r.t. a nominal value of 595 m/s

Continued on next page

Table 2.4 -- continued from previous page

Id	Description
Req-G007	The vehicle shall not exceed a distance of 150 km w.r.t. the nominal TAEM (3σ)
Req-G008	The entry guidance shall be able to deal with dispersions at entry interface of ± 250 m (3σ) in terms of altitude
Req-G009	The entry guidance shall be able to deal with dispersions at entry interface of ± 0.5 deg (3σ) in terms of longitude
Req-G010	The entry guidance shall be able to deal with dispersions at entry interface of ± 0.5 deg (3σ) in terms of latitude
Req-G011	The entry guidance shall be able to deal with dispersions at entry interface of ± 70 m/s (3σ) in terms of velocity modulus
Req-G012	The entry guidance shall be able to deal with dispersions at entry interface of ± 0.5 deg (3σ) in terms of flight-path angle
Req-G013	The entry guidance shall be able to deal with dispersions at entry interface of ± 0.5 deg (3σ) in terms of velocity azimuth angle
Req-G014	The entry guidance shall be able to deal with an uncertainty at entry interface of ± 2.5 kg (3σ) in mass
Req-G015	The entry guidance shall be able to deal with uncertainties of $\pm 20\%$ (3σ) in terms of atmospheric density
Req-G016	The entry guidance shall be able to deal with uncertainties of $\pm 5\%$ (3σ) in terms of lift coefficient
Req-G017	The entry guidance shall be able to deal with uncertainties of $\pm 5\%$ (3σ) in terms of drag coefficient
Req-G018	The entry guidance shall be able to deal with wind profiles up to ± 50 m/s
Req-G019	The AoA modulation shall not be more than $+10$ deg w.r.t. the nominal profile
Req-G020	The AoA modulation shall not be less than -2.5 deg w.r.t. the nominal profile
Req-G021	The AoA-modulation rate shall not be more than $+5$ deg/s w.r.t. the nominal profile
Req-G022	The AoA-modulation rate shall not be less than -5 deg/s w.r.t. the nominal profile
Req-G023	The Bank-angle modulation shall not be more than $+60$ deg
Req-G024	The Bank-angle modulation shall not be less than -60 deg
Req-G025	The Bank-angle rate modulation shall not be more than $+5$ deg/s
Req-G026	The Bank-angle rate modulation shall not be less than -5 deg/s
Req-G027	For the design phase nominal US Standard Atmosphere 1976 shall be used
Req-G028	For the MC campaigns perturbed US Standard Atmosphere

Continued on next page

Table 2.4 -- continued from previous page

Id	Description
	1976 shall be used
Req-G029	For the MC campaigns NRLMSISE00 model for the atmospheric density shall be used
Req-G030	For the design phase gravitational central field with J_2 shall be used
Req-G031	For the MC campaigns gravitational central field with the contribution of the J_2 term shall be used
Req-G032	For the design phase wind profiles shall not be used
Req-G033	For the MC campaigns wind profiles according to HWM07 model shall be used
Req-G034	For the design phase nominal aerodynamic database shall be used
Req-G035	For the MC campaigns perturbed aerodynamic database shall be used
Req-G36	The Entry guidance system shall have two independent guidance schemes based on different techniques, which can run in parallel

Chapter 3

Environment and Aerodynamics modeling

The objective of this chapter is to describe the different modules implemented for the Entry Guidance of SHEFEX-3. These modules will be used to properly characterize terms, which will appear in the formulation of the equations of motion described in Chapter 4 (e.g., the aerodynamic forces). Specifically, we need to characterize the following models:

- Atmosphere;
- Gravity;
- Wind;
- Aerodynamics;

They will be used for the development, the analysis and the verification of the guidance system. Many of them are well-known, and can easily be found in literature. As anticipated in Sec. 2.4 of Chapter 2, there are multiple models which compute in different way the same variables, with different degrees of accuracy and computational cost. A trade-off between these two aspects is the key factor for the selection of the models. The selected models are described in the next sections. Specifically, Sec. 3.1 describes in order of complexity three different atmospheric models. In Sec. 3.2 the gravity models usually employed for the entry mission scenarios are reported, while the wind model used in this work is the subject of Sec. 3.3. Finally, the aerodynamic database is described in Sec. 3.4. The general approach has been to discard models, which were considered too simple (e.g., the exponential atmospheric density), and at the same time, to use the most computationally expensive for the Monte-Carlo campaigns, to verify their impact on the performance of the guidance system.

3.1 Atmosphere

With an atmospheric model we mean the mathematical characterization of a finite (and possibly small) number of parameters representing the thermodynamic quantities we need to know to determine the motion during the entry. For instance, a good knowledge of the atmospheric density allows for a more accurate computation of the dynamic pressure, and the aerodynamic accelerations. The temperature is indirectly used (through the speed of sound), together with the velocity, to know the Mach number, and so on. For characterizing these quantities, it is necessary to build mathematical models to represent them. This yields to the development of analytical and easy-to-handle relationships, such as the exponential model describing the atmospheric density. However, given the uncertainties and the disturbances acting during the real flight phase, more accurate models are needed. Therefore, in the last 50 years, more sophisticated and reliable models have been developed. In the frame of this work, three different models have been considered, and two of them have been implemented for the SHEFEX-3 entry guidance analysis, development and validation. They are reported in order of complexity.

- Exponential Atmosphere;
- US Standard Atmosphere 1976;
- NRLMSISE-00;

3.1.1 Exponential Atmosphere

We refer to the exponential atmosphere (also known as isothermal atmosphere) as a set of analytical expressions which completely characterize all the environmental quantities, like the temperature, and the atmospheric density. This is a standard model used in the design phase of guidance systems (e.g. to compute the reference trajectory). We model the density and the temperature as function of the altitude[27].

3.1.1.1 Density

For the density the following expression can be used,

$$\rho(h) = \rho_0 e^{-\frac{h}{h_s}} \tag{3.1}$$

where ρ_0 is the atmospheric density at sea level, and equal to 1.225 kg/m^3, and h_s is the scale height, which can assume different values (e.g., 7050 m, or 7254.3 m)[28,29]. This profile is of course an approximation, usually used from 0 to 86 km. In the higher region which goes from 86 to 120 km this model computes an atmospheric density which can significantly differ from the real one. As a consequence, this can introduce non-negligible errors in the dynamics of the vehicles. For this reason, in this work it has been chosen to use the US76 Standard Atmosphere 1976, and the NRLMSISE-00 instead.

3.1.1.2 Temperature

Since we are dealing with an isothermal model, the temperature should be constant. However, it can be modeled with the junction of a series of piecewise constant values.

$$T(h) = T_i, \qquad h \in [h_i, h_{i+1}] \tag{3.2}$$

However, this model affects the computation of the speed of sound, and consequently, of the Mach number, which is needed for the computation of the aerodynamic coefficients. Therefore, in many applications this model is not accurate enough.

3.1.2 US Standard 76

A more realistic model is represented by the US Standard Atmosphere 1976, developed by the U.S. Committee on Extension to the Standard Atmosphere[30]. The first model, developed in 1962, was updated in 1966, and then, in its definitive version in 1976. It is based on the assumption of dry, perfect gas mixture. Also in this case all the quantities can be determined with the sole knowledge of the altitude. The components of the atmosphere included in this model and their percentage are shown in Fig. 3.1.

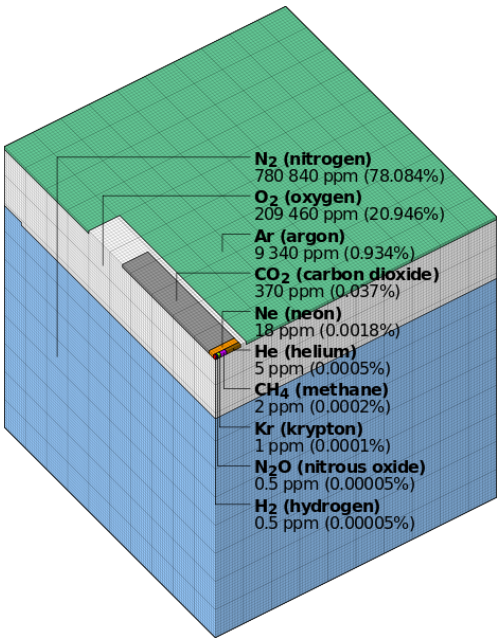


Figure 3.1: Atmosphere composition according to US-76 model - courtesy of Cmglee (creative commons).

3.1.3 Density

The density can be modeled using the ideal gas law,

$$\rho = \frac{pM}{R^* T} = \frac{\sum_{i=0}^{n_s}(n_i M_i)}{N_A} \tag{3.3}$$

where:

- p is the atmospheric pressure;
- R^* is the universal gas constant, equal to 8.3143210^{-3} Nm/kmol K;
- n_i is the number density of atoms of the i^{th} specie;
- M_i is the molecular weight of the i^{th} specie;
- N_A is the Avogadro constant, equal to 6.0221691026 kmol^{-1};
- n_s is the number of the species;

3.1.4 Temperature

The temperature can be computed using the corresponding equations for the ideal gas. The conditions assumed at sea level are

$$\begin{array}{rcl} \rho(0) & = & 1.225 \text{ kg/m}^3 \\ T(0) & = & 288.15 \text{ K} \\ P(0) & = & 101325 \text{ N/m}^2 \\ g(0) & = & 9.80665 \text{ m/s}^2 \end{array} \tag{3.4}$$

These methods are generally implemented for the design phase. However, for the verification of the algorithms, more accurate models need to be used. For this reason, a more complex model, that is, the NRMLSISE-00 has been considered in this work.

3.1.5 NRLMSISE-00 model

NRLMSISE-00 is one of the most accurate atmospheric models developed so far. It can compute not only variables like density and temperature, but the density numbers of the components of the atmosphere. Its use is not restricted to entry scenarios. Indeed, it is usually employed for accurate predictions of satellite orbital decay due to atmospheric drag. This model has also been used by astronomers to calculate the mass of air between telescopes and laser beams to assess the impact of laser guide stars on the non-lasing telescopes. The model is based on the earlier models like MSIS-86[31], but has been updated with more recent satellite drag data. The acronyms **NRL** and **MSIS** stand for the US Naval Research Laboratory and Mass Spectrometer and Incoherent Scatter Radar respectively, which are the two primary data sources for development of earlier

versions of the model, while **E** indicates that the model extends from the ground through exosphere and **00** is the year of release. It is currently the standard for international space research. Since it is one of the most accurate models developed so far, it is of course, one of the most expensive from the computational point of view. The inputs (some of them are optional, but can be provided for more accurate predictions) for this model are

- mission time, meant as *day*, *year*, and *time*;
- altitude h;
- geodetic latitude ϕ;
- longitude θ;
- local apparent solar time;
- 81-day average of $F10.7$ solar flux;
- daily $F10.7$ solar flux for previous day;
- daily magnetic index;

The complete description of the model can be found in the references[25]. Figure 3.2 shows the density and temperature profiles for the different models implemented. In particular, since the NRLMSISE-00 gives the possibility to specify the date of the mission, four different periods of the year (March 21, June 21, September 21, and December 21) have been considered for visualization purposes.

It is interesting to observe that for the density, the exponential model represents, as expected, a good approximation until about 86 km; above 86 km there is a significant difference between this model and the others. Therefore, the NRLMSISE-00 will be a good model to verify the algorithms in presence of strong variations of density w.r.t. the US76. Moreover, the use of NRLMSISE-00 will give us the chance to observe the robustness of the algorithms in presence of perturbations of the Mach number, consequence of different profiles of temperature, which shows stronger variations as a function of the period of the year. Moreover, the solar flux varies and the magnetic activity as well. While US76 will be used for the design phase, a perturbed US76 model, and the NRLMSISE-00 model will be used for the verification phase.

3.2 Gravity Models

Since we are dealing with a non-powered entry, gravity is the most important acceleration acting on our spacecraft, together with the aerodynamic accelerations. As a consequence, a good characterization of the gravitational field acting on the entry vehicle will lead to a more reliable and accurate description of its motion in the atmosphere.

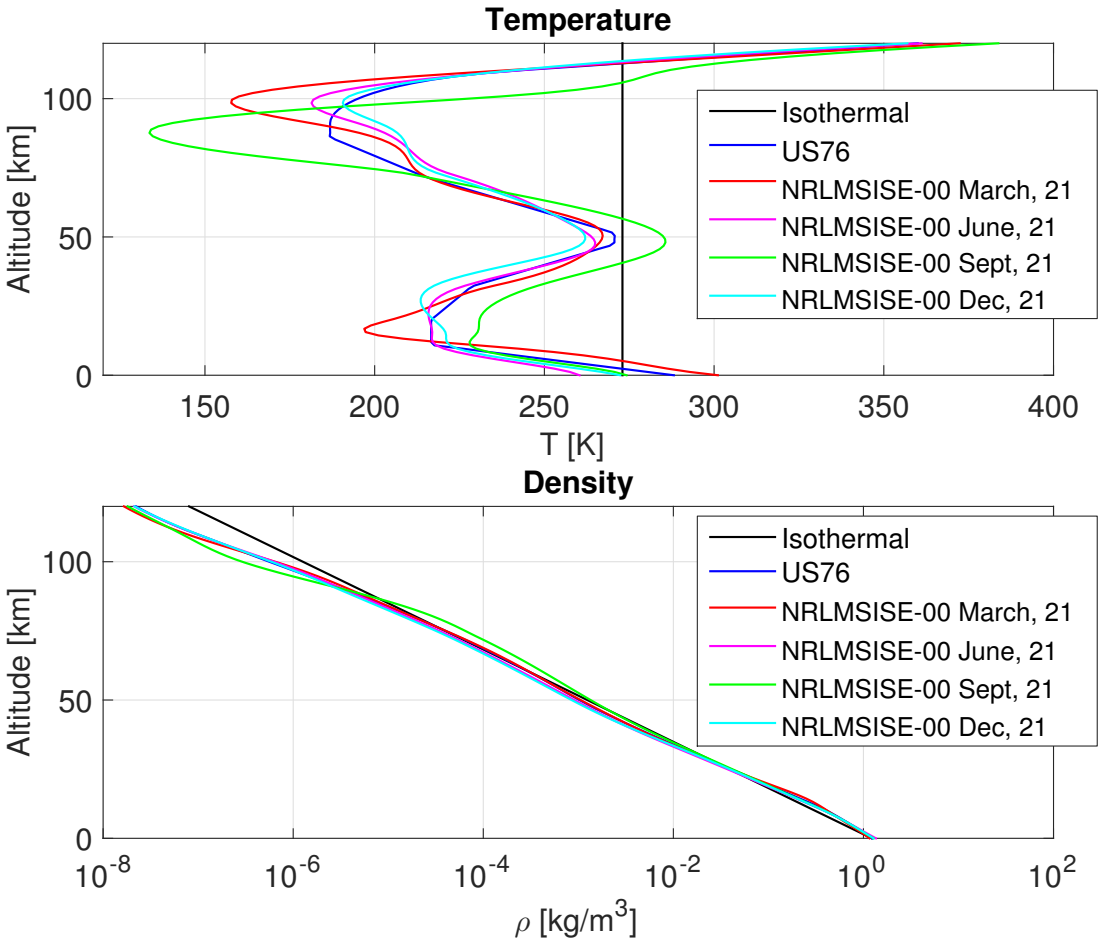


Figure 3.2: Density and Temperature obtained with different atmospheric models.

3.2.1 Central Field

Gravity can be in first approximation modeled as a pure central field generated by the point having the mass of the Earth, and placed in its center. This will lead to the following expression for the gravitational acceleration,

$$g(r) = \frac{\mu_\oplus}{r^2} \tag{3.5}$$

where $\mu_\oplus$ is the gravitational parameter of the Earth, and r is the radial position of the spacecraft, which is (for a spherical Earth) the sum of the altitude and the Earth radius.

$$r = R_\oplus + h \tag{3.6}$$

The reference parameters used for this model are shown in Table 3.1.

Table 3.1: Earth reference parameters[32].

Parameter	Reference Value
$\mu_\oplus$ [km^3/s^2]	$3986004.418 \cdot 10^8$
$R_\oplus$ [km]	6378.137

3.2.2 Gravity model with J_2 term

A more accurate approximation of the gravity field can be obtained if we model better the mass distribution of the Earth. Mathematically, the oblateness of the Earth can be taken into account with the introduction of the zonal harmonic J_2. This correction depends on the radius and the geodetic latitude ϕ, and can be expressed as

$$g(r,\phi) = \frac{\mu_\oplus}{r^2} \cdot \left[1 + \frac{3}{2} \cdot J_2 \cdot \left(\frac{r}{R_\oplus}\right)^2 \left(1 - 3\sin^2\phi\right)\right] \tag{3.7}$$

where the gravitational parameter of the Earth $\mu_\oplus$ has been defined in Section 3.2.1, the zonal coefficient J_2 is $1.0826271 \cdot 10^{-3}$, while the semi-major axis $R_\oplus$ is considered equal to 6378.137 m. Further zonal coefficients can be neglected, as for the mission proposed here, no significant differences were observed w.r.t. approximations having higher order-degree harmonic coefficients (e.g., as in the case of the WGS84).

3.3 Wind

During the real mission, several uncertainties can significantly alter the inflight conditions. One of the most relevant disturbances acting on a non-powered vehicle is the wind. Given its semi-stochastic nature it is an extremely variable disturbance, which depends on the period of the year, the altitude, as well as many other variables. Therefore, to test the guidance system, a mathematical model for the wind must be selected and implemented. The model must be representative from the physical point of view, but, at the same time, easy enough for an efficient implementation. A model having these properties is the Horizontal Wind Model 2007, also referred as HWM07.

3.3.1 Horizontal Wind Model 2007

HWM07 stands for Horizontal Wind Model, released in 2007[24]. It is an empirical model that describes the wind gusts in the terrestrial atmosphere. It represents a trade-off between look-up tables and analytical models. It integrates data coming from the previous versions (HWM87-HWM93) with new measurements sets, obtained with satellite-based observations. From a mathematical point of view, the wind components $\mathbf{u}_i$ can be computed as

$$\begin{aligned}\mathbf{u}_i(t,\theta,\phi,\delta) = &\sum_{n=0}^{N}\sum_{s=0}^{S} C_r^{n,s}\Psi_1^{n,s} - C_i^{n,s}\Psi_2^{n,s} + B_r^{n,s}\Psi_3^{n,s} + B_i^{n,s}\Psi_4^{n,s} \\ &+ \sum_{l=1}^{L}\sum_{n=l}^{N}\sum_{s=0}^{S} C_r^{l,n,s}\Psi_1^{l,n,s} - C_i^{l,n,s}\Psi_2^{l,n,s} \\ &+ B_r^{l,n,s}\Psi_3^{l,n,s} + B_i^{l,n,s}\Psi_4^{l,n,s} + \sum_{m=1}^{M}\sum_{n=m}^{N}\sum_{s=0}^{S} C_r^{m,n,s}\Psi_1^{l,n,s} \\ &- C_i^{m,n,s}\Psi_2^{m,n,s} + B_r^{m,n,s}\Psi_3^{m,n,s} + B_i^{m,n,s}\Psi_4^{m,n,s}\end{aligned} \tag{3.8}$$

where:

- t is the time;
- θ is the geodetic longitude;
- ϕ is the geodetic latitude;
- δ is the local solar time;
- s, l, m, n are the seasonal, local time, longitudinal and latitudinal spectral wave numbers;
- $C_{r,i}^{n,s}$, $B_{r,i}^{n,s}$, $C_{r,i}^{l,n,s}$, $B_{r,i}^{m,n,s}$ are the spectral coefficients, estimated experimentally;
- S, L, M, N are the maximum spectral orders, that, for the HWM07 model, are equal to 2, 3, 2, and 8 respectively;
- Ψ_1,Ψ_2,Ψ_3,Ψ_4 are the modulated basis function of the Fourier series and computed as follows

$$\begin{aligned}\Psi_1^{g,n,s}(\beta,\phi,t) &= -\frac{\cos g\beta}{\sqrt{n(n+1)}}\frac{d\bar{P}_n^g(\phi)}{d\phi}\left[\cos(st)+\sin(st)\right] \\ \Psi_2^{g,n,s}(\beta,\phi,t) &= \frac{\sin g\beta}{\sqrt{n(n+1)}}\frac{d\bar{P}_n^g(\phi)}{d\phi}\left[\cos(st)+\sin(st)\right] \\ \Psi_3^{g,n,s}(\beta,\phi,t) &= -\frac{\sin g\beta}{\sqrt{n(n+1)}}\frac{\bar{P}_n^g(\phi)}{\cos\phi}\left[\cos(st)+\sin(st)\right] \\ \Psi_4^{g,n,s}(\beta,\phi,t) &= -\frac{\cos g\beta}{\sqrt{n(n+1)}}\frac{\bar{P}_n^g(\phi)}{\cos\phi}\left[\cos(st)+\sin(st)\right]\end{aligned} \tag{3.9}$$

$\bar{P}_n^g(\phi)$ are the normalized associate Legendre Polynomials, g can be either m or l, while β can be δ or θ. The wind velocity vector can then be computed with the use of

$$\mathbf{U}(t,\theta,\phi,\delta) = \sum_j \gamma_j \mathbf{u}_j(t,\theta,\phi,\delta) \tag{3.10}$$

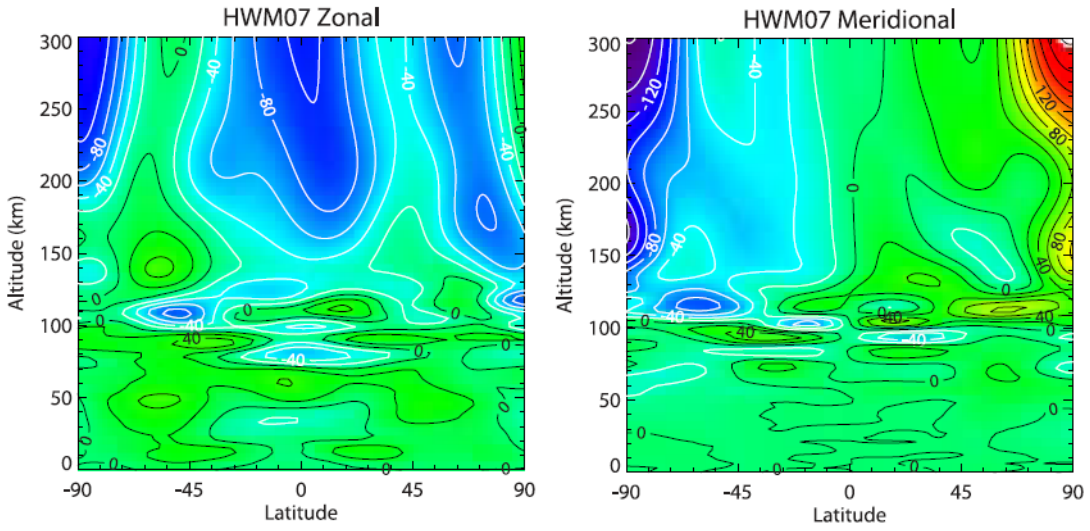


Figure 3.3: Zonal and meridional wind maps computed with HWM07[24].

with the components of $\mathbf{u}_j$ that can be computed using the 3.8, and the terms γ_i are some properly defined weighting terms. Figure 3.3 shows some examples of wind maps computed with the use of the HWM07 model, while Fig. 3.4 shows a comparison between the models HWM07 and HWM93 in the period November-February for different local times.

The wind profiles are plotted together with experimental data obtained with the Wind Imaging Interferometer (WINDII) satellite[24]. Indeed, while the HWM93 overestimates the wind, the predictions made with the HWM07 model agree with the observations. In terms of magnitude of perturbations, it is possible to observe a maximum perturbation of about 50 m/s. The HWM07 model will be used to take the wind into account during the Monte Carlo validation campaign. How the wind has been integrated within the equations of motion will be explained in Chapter 4.

3.4 Aerodynamics

As stated in Section 3.2, aerodynamic accelerations are, together with the gravity, the most important accelerations to be taken into account during the entry. It is thus essential to implement a good mathematical representation of the aerodynamics associated with the SHEFEX-3 vehicle. This is characterized by the coefficients of lift C_L, and drag C_D. Moreover, since the vehicle must be trimmed to guarantee its controllability, the flap deflection τ, which realizes the trim conditions, needs to be computed. In this section, all these quantities will be computed using the data provided by DLR aerodynamics experts. Starting from the body-reference coefficients C_x and C_z, we will build the aerodynamic database represented by the lift coefficients C_L, the

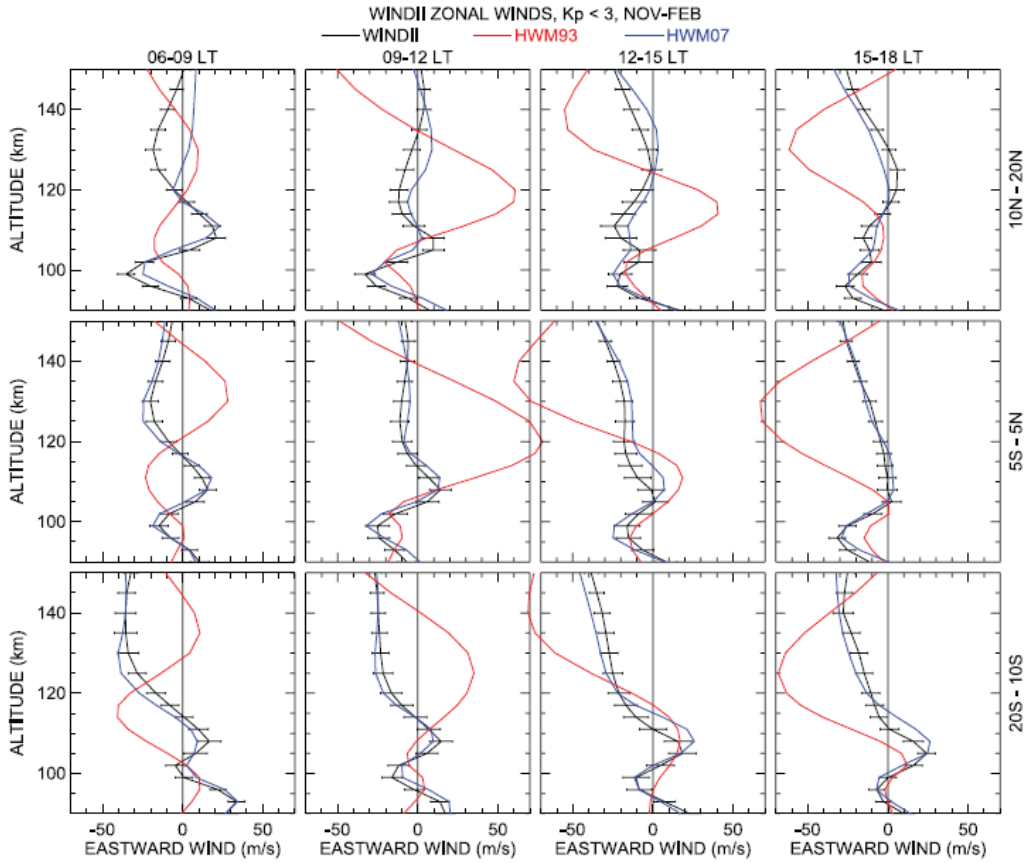


Figure 3.4: Comparison of wind models between 0 and 300 km with models HWM07 and HWM93[24].

drag coefficient C_D, and the flap deflection τ. These coefficient take the friction into account via heuristic formulas[1], and a post-processing procedure for generating smooth data will be explained.

3.4.1 SHEFEX-3 basic aerodynamics

The post-processing applied to the aerodynamics database is explained below. The data has been generated by solving Euler equations, which do not include friction. Friction coefficients are computed by using heuristic formulas, and added to the inviscid coefficients. Finally, the results have been verified with Navier-Stokes equations, and validated with wind-tunnel tests[26]. Since the data is provided for a finite number

[1]The author apologizes for the lack of proper references in this case. The aerodynamic database has been provided by DLR aerodynamics experts. The results, the models used, and the corrections via heuristic formulas have been described during several meetings and presentations in the DLR institutes, and therefore are not publically available.

of angles of attack, Mach numbers, and altitude, it needs to be post-processed. The post-processing of the aerodynamics for SHEFEX-3 has the following objectives:

- Derivation of the trim conditions for each angle-of-attack and Mach number condition;
- Computation of lift and drag coefficients associated with each trim condition;
- Smoothing the aerodynamics data with the use of cubic splines;
- Generation of smooth maps of friction coefficient w.r.t. altitude, Mach number and flap deflections via least-squares fitting;
- Integration of processed friction with the aerodynamics database;

3.4.2 Derivation of the trim conditions

To control the vehicle during the entry phase, the trim conditions need to be derived. Nominally, the trimming conditions are realized when the total moments (or in equivalent form, their non-dimensional coefficients) acting on the vehicle are equal to 0.

$$\sum M_i = 0 \tag{3.11}$$

where M_i represents the i^{th} moment component acting on the vehicle. The aerodynamic database is organized as follows. For different values of Mach numbers, angles of attack and flap deflection τ, the following data are given.

- Total pitch moment coefficient $C_{M,O}$
- Aerodynamic force coefficient of the body w.r.t. the Z axis $C_{Z,B}$;
- Aerodynamic force coefficient of the body w.r.t. the X axis $C_{X,B}$;
- Aerodynamic force coefficient of the flap w.r.t. the Z axis $C_{Z,F}$;
- Aerodynamic force coefficient of the flap w.r.t. the X axis $C_{X,F}$;

The database is computed w.r.t. the origin of the body reference frame, placed in the nose of the vehicle. We are interested in the equilibrium condition around the CoM of the vehicle, which is located at 55% of the length of the vehicle.

$$\begin{array}{rcl} CoM_X & = & 55\% \\ CoM_Z & = & 0\% \end{array} \tag{3.12}$$

The first step is the computation, for each condition provided, of the C_M around the CoM. In terms of coefficients, the total pitching moment around the CoM is

$$C_{M,CoM} = C_{M,O} - (C_{X,B} + C_{X,F})\, CoM_Z + (C_{Z,B} + C_{Z,F})\, CoM_X \tag{3.13}$$

where $C_{M,CoM}$ is the pitching moment coefficient w.r.t. the CoM. Since the center of mass lies on the axis, CoM_Z is equal to 0, and there is no contribution of the coefficients $C_{X,B}$ and $C_{X,F}$.

With Eq. 3.13 it is possible to compute the total pitch coefficient for each angle of attack, flap deflection and Mach number. Specifically, for each α and Mach number we can find the two flap deflections which include the trim conditions, that is the flap deflections where the $C_{M,CoM}$ changes its sign.

$$\begin{aligned} \tau^+ &: C_{M,CoM}(\tau^+) > 0 \\ \tau^- &: C_{M,CoM}(\tau^-) < 0 \end{aligned} \tag{3.14}$$

The flap deflection, which realizes the trim condition is then computed interpolating linearly between these two values. So we can write

$$C_{M,CoM}(\tau) = a\tau + b, \quad \tau \in [\tau^-, \tau^+] \tag{3.15}$$

where

$$\begin{aligned} a &= \frac{C_{M,CoM}(\tau^+) - C_{M,CoM}(\tau^-)}{\tau^+ - \tau^-} \\ b &= C_{M,CoM}(\tau^+) - a\tau^+ \end{aligned} \tag{3.16}$$

and the flap deflection, which generates the trim condition is

$$\tau_{trim} = -\frac{b}{a} \tag{3.17}$$

Hence, it is possible to compute the flap deflection such that

$$C_{M,CoM} = 0 \tag{3.18}$$

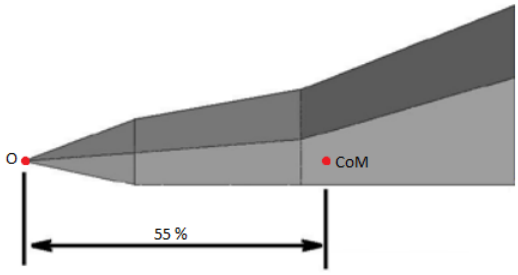


Figure 3.5: Reference Points for Pitch Equilibrium.

3.4.3 Computation of lift ad drag coefficients C_L and C_D

Once the flap deflections for all the conditions are computed, the lift and drag coefficients can be computed as well. Assuming that the side-slip angle is zero, the analysis can be reduced to the longitudinal plane, i.e., the X-Z plane, oriented as shown in Fig. 3.7.

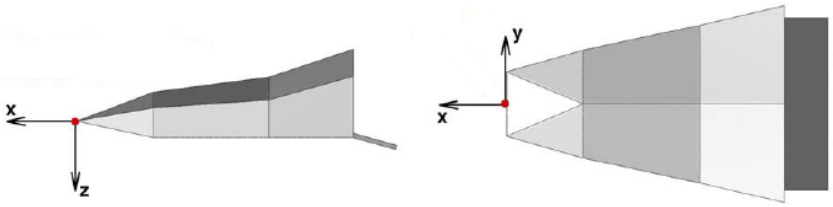


Figure 3.6: SHEFEX-3 CFD Reference Frame.

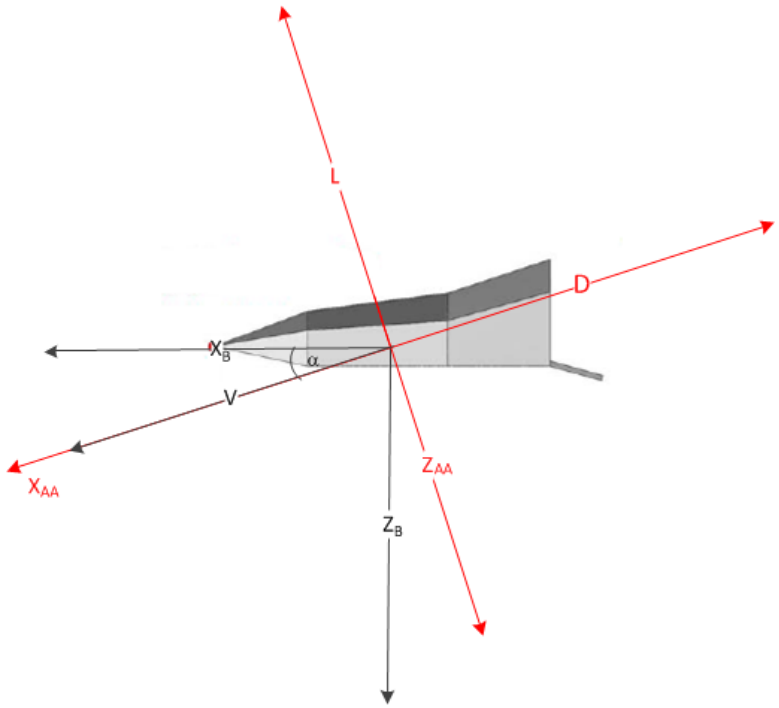


Figure 3.7: SHEFEX-3 Aerodynamic and Body Reference Frame.

For a given α, the C_L and C_D can be computed transforming the coefficients C_X and C_Z in the aerodynamic reference frame, where the drag is parallel to the velocity vector, but opposite in direction, and the lift is perpendicular to drag and directed upwards. It

is then possible, for each angle of attack, Mach number and the flap deflections τ^+ and τ^- defined in the previous section, to compute the lift and drag coefficients as

$$\begin{array}{ll} C_L^+ = -C_Z\cos(\alpha) + C_X\sin(\alpha), & \tau = \tau^+ \\ C_D^+ = -C_Z\sin(\alpha) - C_X\cos(\alpha), & \tau = \tau^+ \\ C_L^- = -C_Z\cos(\alpha) + C_X\sin(\alpha), & \tau = \tau^- \\ C_D^- = -C_Z\sin(\alpha) - C_X\cos(\alpha), & \tau = \tau^- \end{array} \tag{3.19}$$

Then, also in this case, we can operate a linear interpolation to compute the aerodynamics coefficients corresponding to the trim conditions. We can define a_L^t, b_L^t, a_D^t, b_D^t as

$$\begin{array}{rcl} a_L^t & = & \frac{C_L^+ - C_L^-}{\tau^+ - \tau^-} \\ b_L^t & = & C_L^- \\ a_D^t & = & \frac{C_D^+ - C_D^-}{\tau^+ - \tau^-} \\ b_D^t & = & C_D^- \end{array} \tag{3.20}$$

and the lift and drag coefficients are

$$\begin{array}{l} C_{L,trim} = a_L^t\left(\tau_{trim} - \tau^-\right) + b_L^t \\ C_{D,trim} = a_D^t\left(\tau_{trim} - \tau^-\right) + b_D^t \end{array} \tag{3.21}$$

Figures 3.8-3.10 show the lift coefficients, the drag coefficients and the flap deflections obtained for SHEFEX-3 based on the trim conditions, without any friction. The results are then linearly interpolated over the entire range of M and α.

It is noted that the vehicle is trimmable over the considered ranges of α and M, as the flap deflections for SHEFEX-3 can continuously vary from -15 to 15 deg, while the maximum deflection found in the region of interest is about 11 deg.

3.4.4 Smoothing of the aerodynamics database

Once the trim conditions have been generated, a further step for processing the aerodynamic data is done. To speed up the computation of the guided reference trajectories, a cubic spline interpolation process has been applied to each of the sets obtained in the previous step. This allows to have continuously differentiable data over the range of interest. The obtained aerodynamic database computed so far does not take any friction effect into account. This will be the topic of the next section.

3.4.5 Processing of Friction Data

The coefficients previously computed do not take friction into account, as they have been computed using ideal solutions of the flow-field. To include friction, a second database, providing the friction C_f as a function of h, M, and τ, is given. This database has been computed using heuristic formulas, and validated using local numerical solutions of the Navier-Stokes equations.
Also in this case, a smoothing process is needed. This will be done using a 3-D polynomial

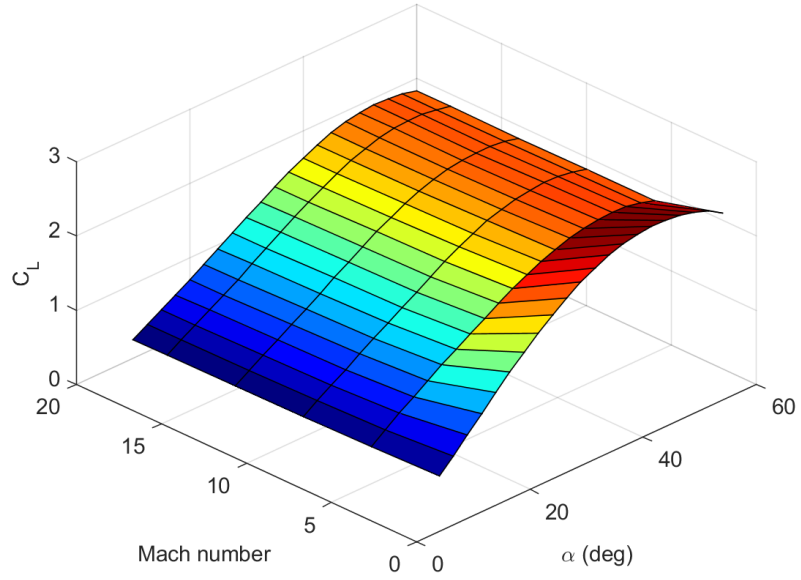


Figure 3.8: Lift coefficient without friction - Linear interpolation.

expression. The proposed template is a complete quadratic 3-D function, having the following form:

$$C_f(h, M, \tau) = \sum_{i=0}^{m_t} \sum_{j=0}^{n_t} \sum_{k=0}^{p_t} a_{i,j,k} h^i M^j \tau^k \tag{3.22}$$

The next step is then to compute the coefficients $a_{i,j,k}$ for the polynomial expression in Eq. 3.22. To find them, a least-squares problem has been defined. The cost function is defined as the sum of squares of the difference between the friction coefficients and the continuous function evaluated in the corresponding points.

$$J = \sum_{m=0}^{m_t} \sum_{n=0}^{n_t} \sum_{p=0}^{p_t} \left[C_f(h_m, M_n, \tau_p) - \sum_{i=0}^{m_t} \sum_{j=0}^{n_t} \sum_{k=0}^{p_t} a_{i,j,k} h_m^i M_n^j \tau_p^k \right]^2 \tag{3.23}$$

The indices m_t, n_t, and p_t are equal to 2, hence 27 coefficients have been determined with this procedure.

To give an example, Figs. 3.11 and 3.12 show the coefficients w.r.t. altitude and Mach number, for two given values of the flap deflection τ, which are -7 deg and $+7$ deg, respectively.

Even if at first glance they look similar, they have been computed in correspondence of different Mach numbers, according to the information provided by DLR aerodynamics

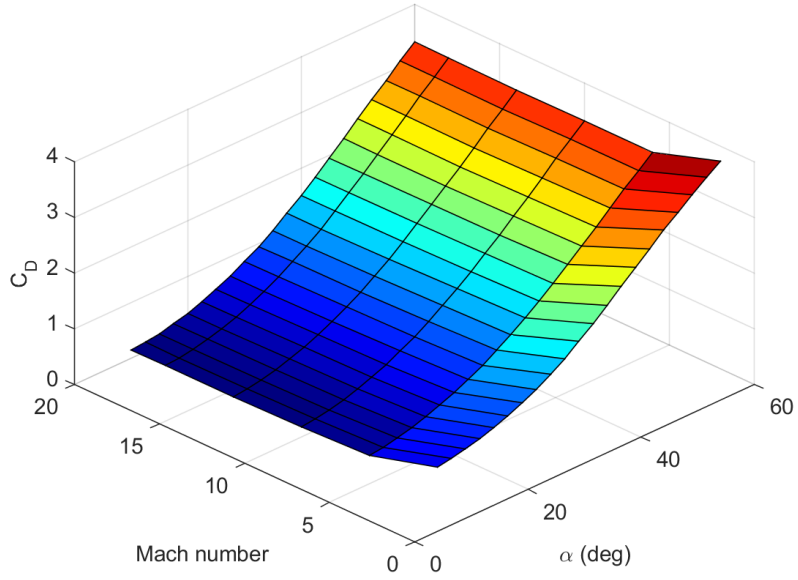


Figure 3.9: Drag coefficient without friction - Linear interpolation.

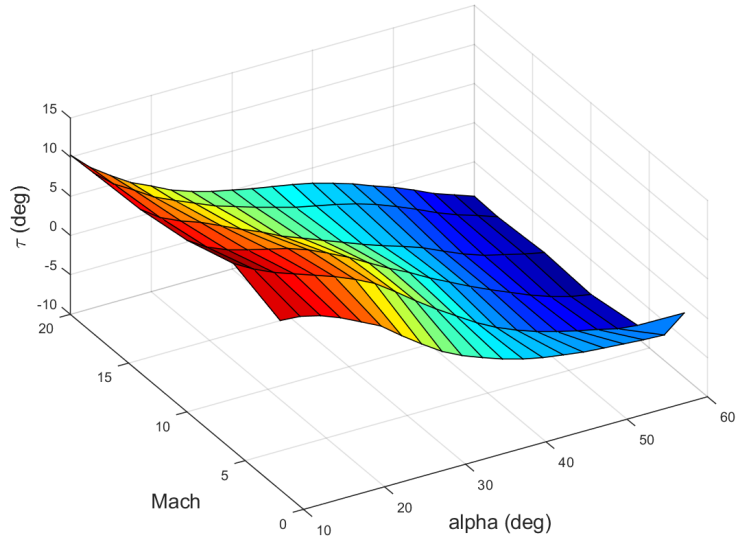


Figure 3.10: Flap Deflection - Linear interpolation.

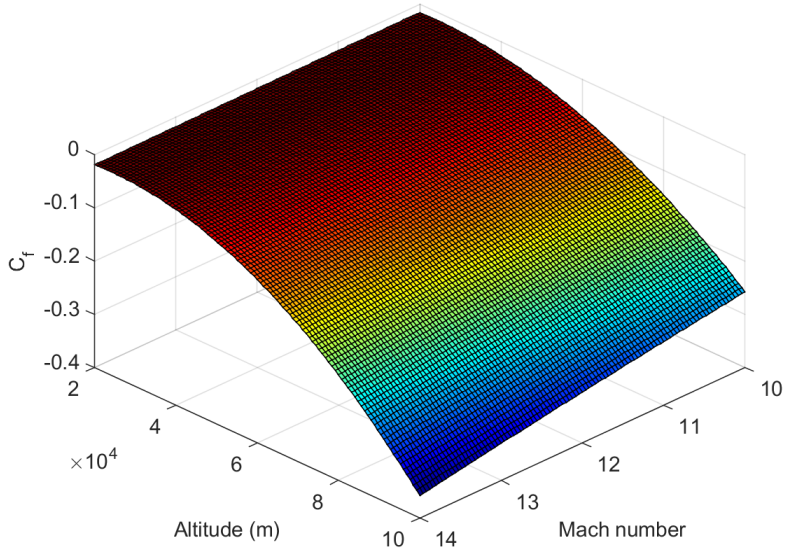


Figure 3.11: Friction Coefficients for a flap deflection $\tau = -7$ deg.

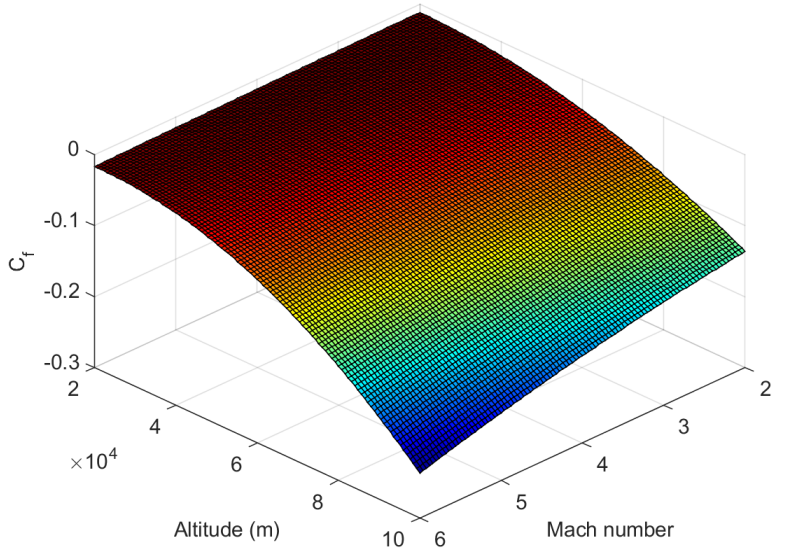


Figure 3.12: Friction Coefficients for a flap deflection $\tau = 7$ deg.

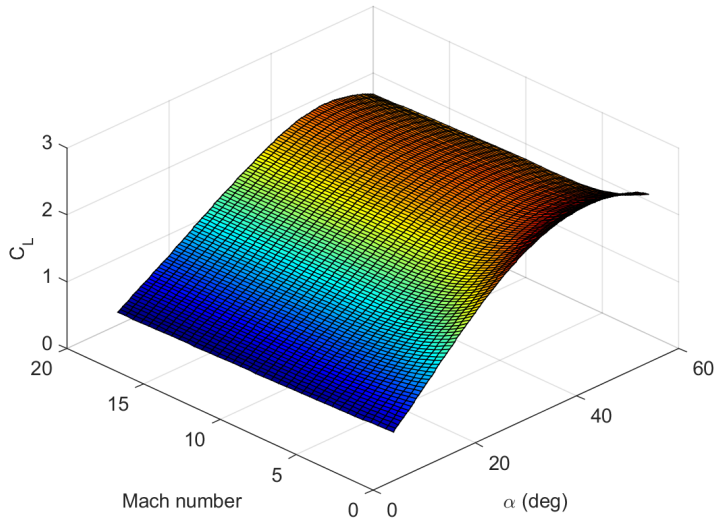


Figure 3.13: Lift coefficient with friction - Spline interpolation.

experts. In Fig. 3.11 the range of Mach numbers considered is between 10 and 14, and the flap deflection is equal to -7 deg. For instance, for $M = 14$ and $h = 100$ km, C_f is $\cong -0.38$. In Fig. 3.12 the flap deflection considered is 7 deg. For $M = 6$ and $h = 100$ km C_f is $\cong -0.23$.

3.4.6 Integration of friction within the aerodynamics database

Once the friction coefficient has been computed, it is possible to correct the aerodynamic coefficients as follows:

$$\begin{aligned} C_L &= C_L - C_f \cos(\alpha) \\ C_D &= C_D + C_f \sin(\alpha) \end{aligned} \tag{3.24}$$

Figures 3.13 and 3.14 show the final version of the aerodynamic database implemented for SHEFEX-3, while Fig. 3.15 reports the C_L/C_D ratio obtained for SHEFEX-3. We can observe a C_L/C_D ratio varying from about 0.8 at high angles of attack to 1.8 at low angles of attack. Smaller variations are observed for a given angle of attack when the Mach number varies. This database has been used for the computation of the entry trajectories for SHEFEX-3. The use of a smooth database has a double advantage. It makes the trajectory optimization more accurate, because all the discontinuities associated with the use of linear interpolation techniques disappear, and more advanced techniques for the differentiation, such as the complex-step approach [14], or the dual number-based approach [10,33,34] can be used. These techniques provide more accurate derivatives, as we will see in Chapter 7.

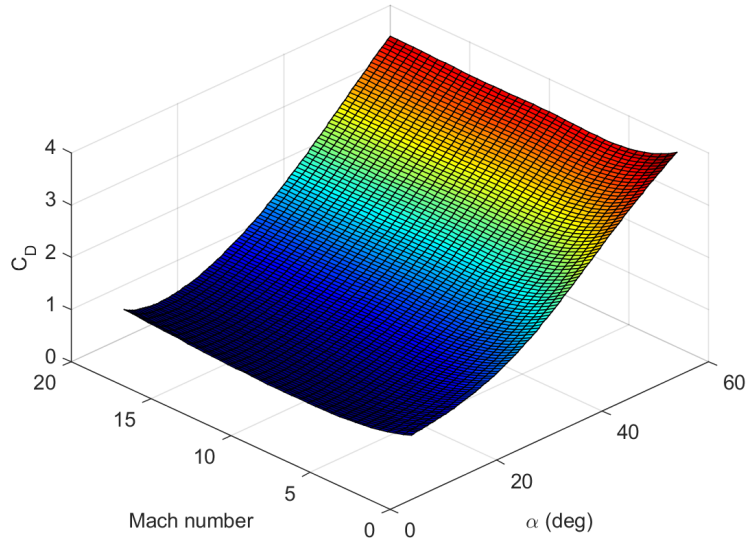


Figure 3.14: Drag coefficient with friction - Spline interpolation.

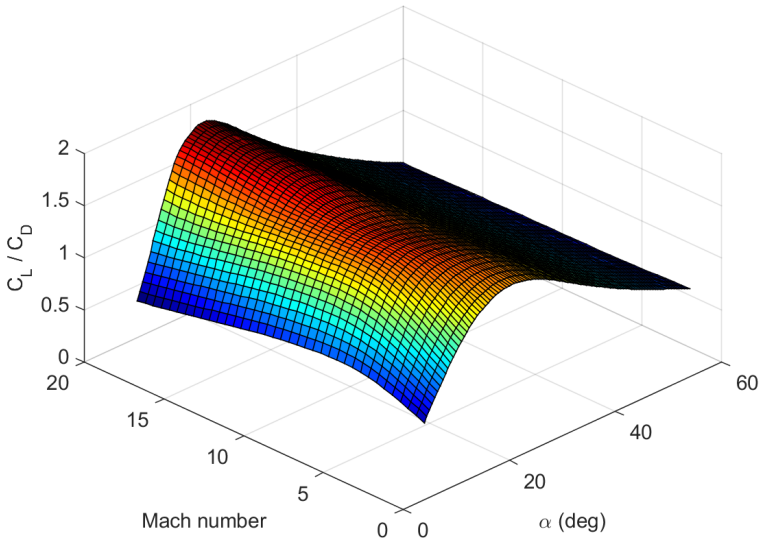


Figure 3.15: SHEFEX-3 Lift-to-Drag ratio.

Chapter 4

Entry Flight Mechanics

This chapter contains a brief description of the reference frames and the coordinate systems used for the formulation of the equations of the motion. It also includes the differential models implemented for the realization of the simulations performed in this work. The independent variable for the system of differential equations can be either time or energy. The corresponding models are mathematically different, but perfectly equivalent from a physical point of view. Both of them have been used for the Earth entry problem, but they can be potentially used for the computation of the entry of any trajectory of a non-powered vehicle in any atmosphere of a celestial body, like Mars. The chapter is structured as follows: in Sec. 4.1 the several reference frames used in this work are described. Some of them are required for the formulation of the equations of motion (e.g., the Trajectory reference frame), and others (like for instance the Downrange-Crossrange-Altitude) are introduced to characterize the performance of the guidance systems. In Sec. 4.2 the equations of motion are reported. The difference among the different models are described, and their use in the work is explained. Finally, in Sec. 4.3 the method to integrate the wind in the equations of motion is explained in detail.

4.1 Reference Frames

The first thing we need to do is to define the reference frames associated with the formulation of the equations of motion. In three-dimensional space, the motion of an object is characterized by the knowledge of a moving reference frame (for instance, the body reference frame, which is fixed to the vehicle) with respect to an external reference frame. This external reference frame can be inertial or non-inertial. For the terrestrial entry problems, usually the position of the vehicle is expressed in polar coordinates w.r.t. the Earth-centered, Earth-Fixed (ECEF) reference frame.

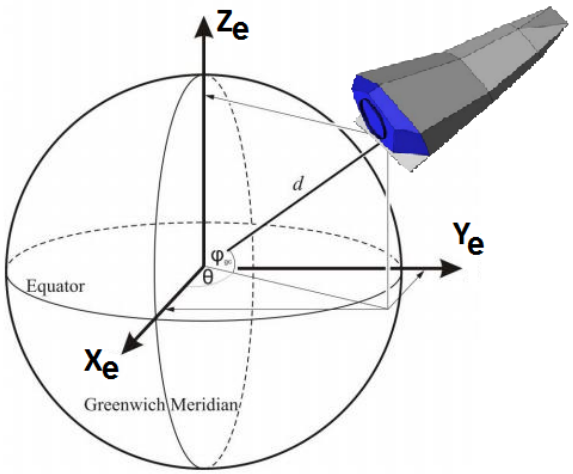


Figure 4.1: Earth-Centered Earth-Fixed reference frame.

4.1.1 Earth Centered Reference Frame

ECEF (Earth-Centered Earth-Fixed Reference Frame) is one of the most common reference frames for terrestrial applications[35]. It is a Cartesian reference frame having the X_e axis lying in the equatorial plane and pointing in the direction of the projection of the Greenwich meridian on the equator. The Z_e axis is parallel to the axis of rotation of the Earth. The axis Y_e is defined such that the reference frame is right handed. This reference frame is plotted in Fig. 4.1. The ECEF will represent the reference frame for the description of the motion of the vehicle during entry.
It is important to stress that this is not the only valid choice, as other reference frames, (e.g., inertial reference frames) may be used. To complete the description of the motion other three reference frames need to be defined. They are the body reference frame, the trajectory reference frame, and the aerodynamic reference frame. Moreover, the meridional-zonal-up, the north-east-down reference, and the local-vertical, local-horizontal reference frames, useful for the integration of wind in the equations of motion, are described. They are defined in the next sections.

4.1.2 Body Reference Frame

The body reference frame is centered in the CoM of the vehicle. For SHEFEX-3, it is defined as having the X_b axis along the longitudinal axis of the vehicle, the Y_b axis points toward the left side of the vehicle, and the Z_b axis defined by the cross product between the axes X_b and Y_b (Fig. 4.2). This reference frame can be linked to the aerodynamic reference frame using the angle of attack and the bank angle. The angle of attack is defined as the angle between the X_b axis and the projection of the velocity vector in the longitudinal plane of the vehicle. The bank angle σ is defined as the angle between

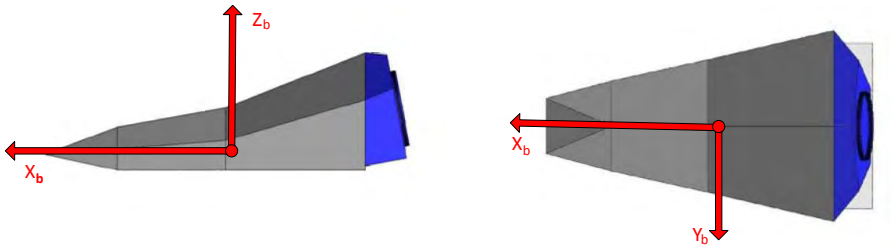


Figure 4.2: SHEFEX-3 Body reference frame.

the plane formed by the velocity vector and the lift vector and the vertical plane in the local horizontal frame that contains the velocity vector. They are represented in Fig. 4.4.

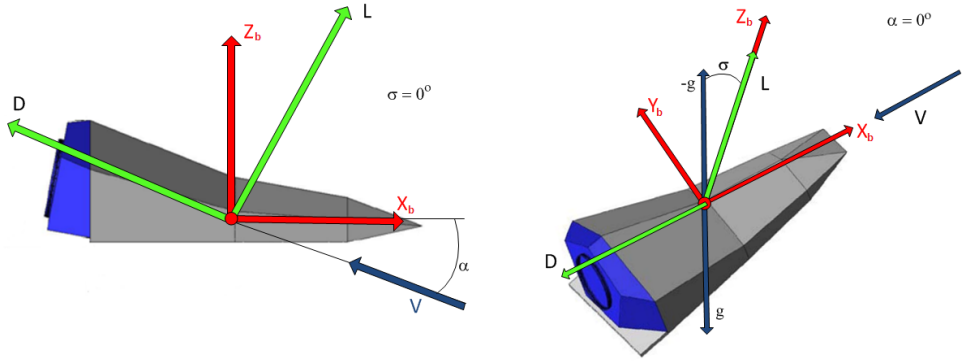


Figure 4.3: angle of attack and bank angle definitions.

4.1.3 SHEFEX-3 Aerodynamic Reference Frame

The aerodynamic reference frame is centered in the CoM of the vehicle. It is necessary to distinguish between two types of aerodynamic reference frames. The groundspeed-based aerodynamic reference frame (AG) refers to the velocity vector evaluated w.r.t. the ground. The X_{ag} axis is parallel to the drag acceleration D_g and opposite in direction, and w.r.t. the body X_b axis of the vehicle is inclined by the groundspeed-based angle of attack α_g, while the axis Z_{ag} is parallel to the lift acceleration L_g, and opposite in direction too. Y_{ag} is given by the cross product between Z_{ag} and X_{ag}. The formal definition remains the same for the airspeed aerodynamic reference frame, formed by the axes X_{aa}, and Z_{aa}, which are parallel and opposite to the drag and lift accelerations D_a

and L_a. The difference between the two reference frames is given by the wind velocity vector V_w, which modifies the velocity that the vehicle effectively experiences. In case there is no wind, the two reference frames coincide. However in general this is not the case, and this difference needs to be taken into account for correctly computing the aerodynamic forces, as we will see in Sec. 4.3.

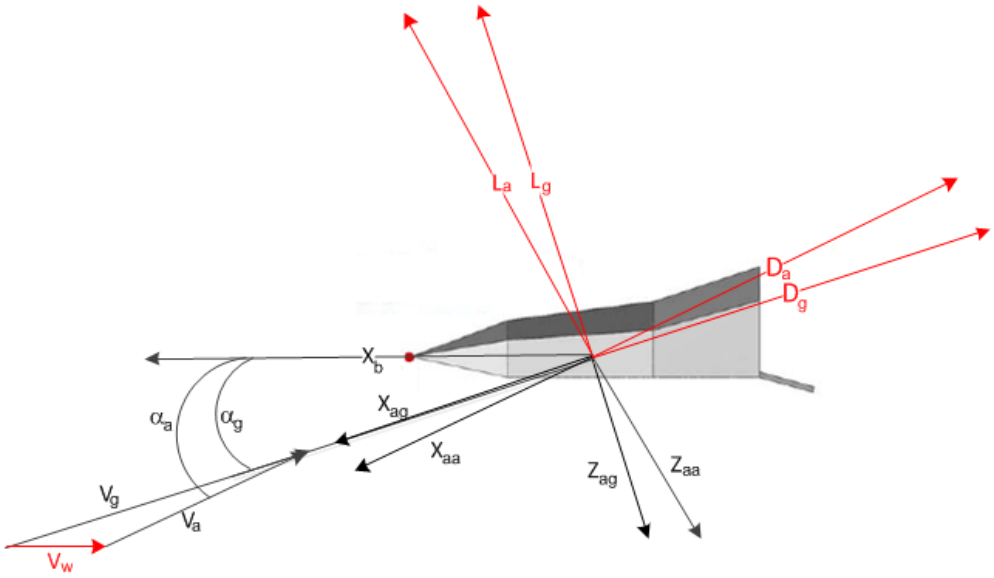


Figure 4.4: angle of attack and bank angle definitions.

4.1.4 North-East-Down Reference Frame

The North-East-Down reference frame is a common reference frame in aeronautical and space applications. It has its origin in the center of mass of the vehicle. As the name suggests, the North and the East axes are directed towards the local North and East, while the Down axis points towards the center of the central body. An example of the NED reference frame is depicted in Fig. 4.5.

4.1.5 Meridian-Zonal-Up Reference Frame

A reference frame similar to the North-East-Down reference frame is the meridional-zonal-up reference frame, which is used by the Horizontal Wind Model 07[24]. The meridional axis points towards the local South, while the zonal axis points again towards the local East. The third axis, here called up, points in opposite direction w.r.t. the center of the central body. The MZU reference frame is depicted in Fig. 4.6.

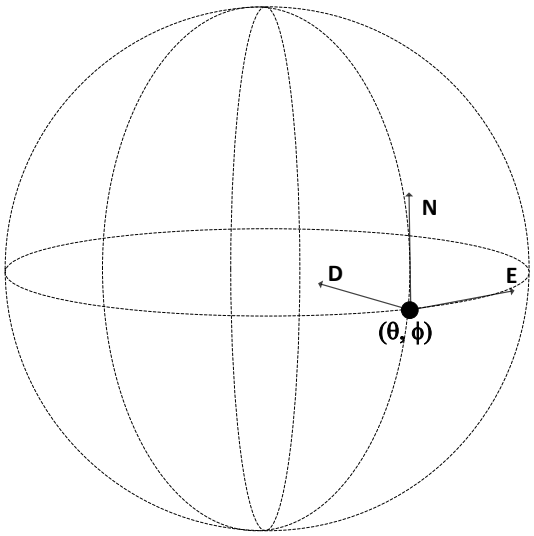


Figure 4.5: North-East-Down reference frame.

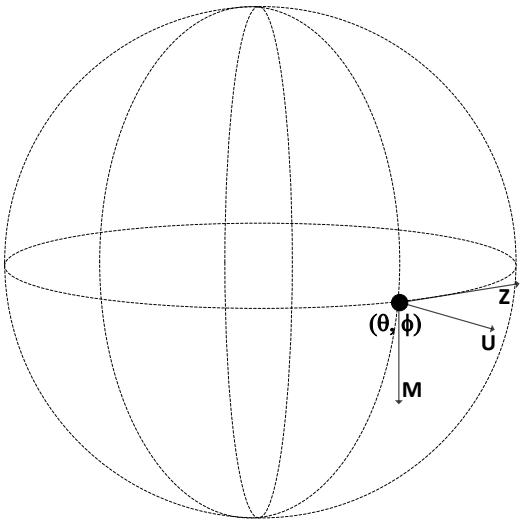


Figure 4.6: Meridional-Zonal-Up reference frame.

4.1.6 Downrange-Crossrange-Altitude Reference Frame

The downrange-crossrange-altitude (DCA) is not used for the formulation of the equations of motion, but is introduced to better evaluate the performance of the system in terms of groundtrack and final dispersions. Theoretically, the downrange is defined as the distance measured as arc over a circle connecting the initial and final nominal points.

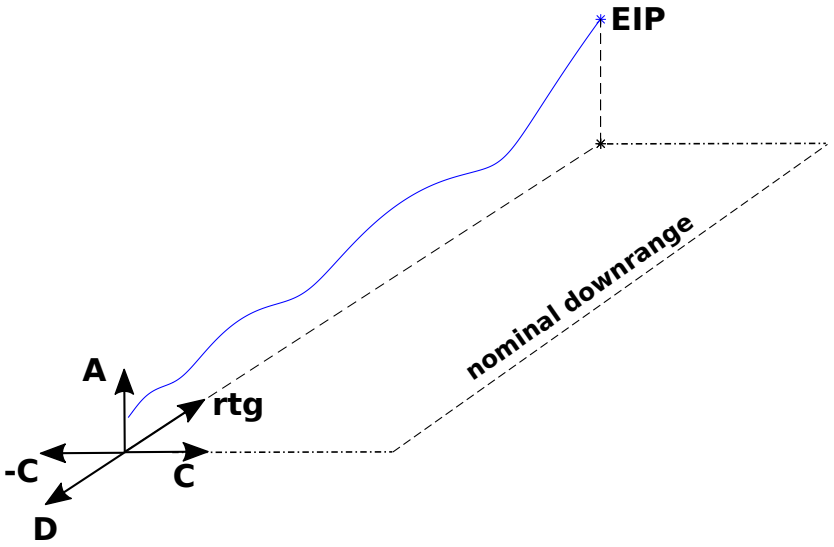


Figure 4.7: Downrange-Crossrange-Altitude reference frame.

However in this context we will define it as pointing from the nominal initial position towards the nominal terminal position. The origin is centered in the terminal position. The crossrange is measured along a line, which direction is 90 deg counter-clockwise w.r.t. the downrange axis and orthogonal to the gravity acceleration vector.

With these definitions, it is possible to measure the offset from the nominal target in terms of range-to-go, given by the downrange, and having opposite sign. For consistency, the lateral error will be measured along the negative direction of crossrange, to have a right-hand reference frame. In nominal conditions, the offset along the crossrange and the downrange directions is zero, and the EIP has zero crossrange offset by definition. The altitude is measured along the axis normal to the plane defined by crossrange and downrange, positive if opposite to the gravity acceleration vector, and measured w.r.t. the mean radius of the Earth. A representation of the DCA reference frame is depicted in Fig. 4.7.

4.1.7 Local Vertical Local Horizontal Reference Frame

The LVLH reference frame is useful for the integration of the wind in the equations of motion. It is centered in the vehicle's center of mass, and has the Z_{lvlh} axis pointing towards the central body, and the X_{lvlh} axis parallel to the velocity vector. The Y_{lvlh} axis is given by the cross product between Z_{lvlh} and X_{lvlh}. A representation of the LVLH reference frame is depicted in Fig. 4.8.

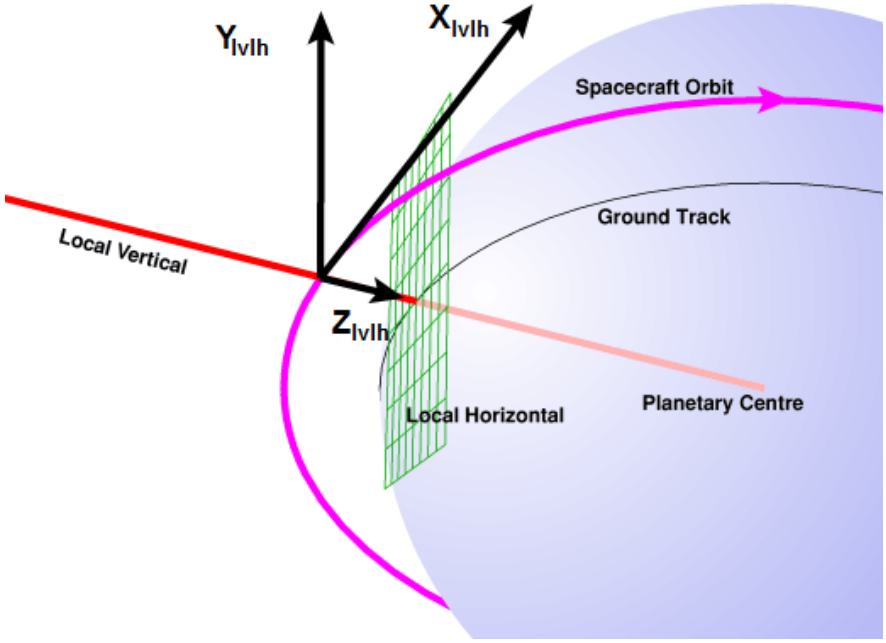


Figure 4.8: Local-Vertical Local Horizontal reference frame.

4.1.8 Trajectory Reference Frame

The equations of motion typically provide a description of the three components of the position and the velocity. However, for the entry problem, the equations of motion are formulated expressing the components of the position as h, (or alternatively the radial position r), geocentric longitude θ and latitude ϕ. The velocity is expressed as magnitude V, flight-path angle γ and velocity azimuth angle ψ. The flight-path angle is the angle between the velocity vector and the local horizontal plane; the north-velocity azimuth angle is defined as the angle between a reference direction, (in this case the North), and the projection of the velocity vector onto the local horizontal plane. The trajectory reference frame has the same X_{tg} axis as the aerodynamic reference frame, while the other two axis depend on the angles α_g, σ_g and β_g. The trajectory reference frame is depicted in Fig. 4.9, together with the body reference frame and the aerodynamic reference frame[36]. As for the aerodynamic reference frame, we can have the groundspeed-based trajectory (tg) reference frame, or the airspeed-based trajectory (ag) reference frame, associated with their respective angles. For instance, if we refer to the TG reference frame, the axes can be computed by transforming the body reference frame as follows[36]

$$\begin{bmatrix} \mathbf{X}_{tg} \\ \mathbf{Y}_{tg} \\ \mathbf{Z}_{tg} \end{bmatrix} = \mathbf{T}_b^{tg} \begin{bmatrix} \mathbf{X}_b \\ \mathbf{Y}_b \\ \mathbf{Z}_b \end{bmatrix} \tag{4.1}$$

where the rotation matrix $\mathbf{T}_b^{tg}$ is

$$\mathbf{T}_b^{tg} = \mathbf{T}_2(\alpha_g)\mathbf{T}_3(-\beta_g)\mathbf{T}_1(-\sigma_g) \tag{4.2}$$

This aspect will be deeply explained in Sec. 4.3. All the equations of motion formulated

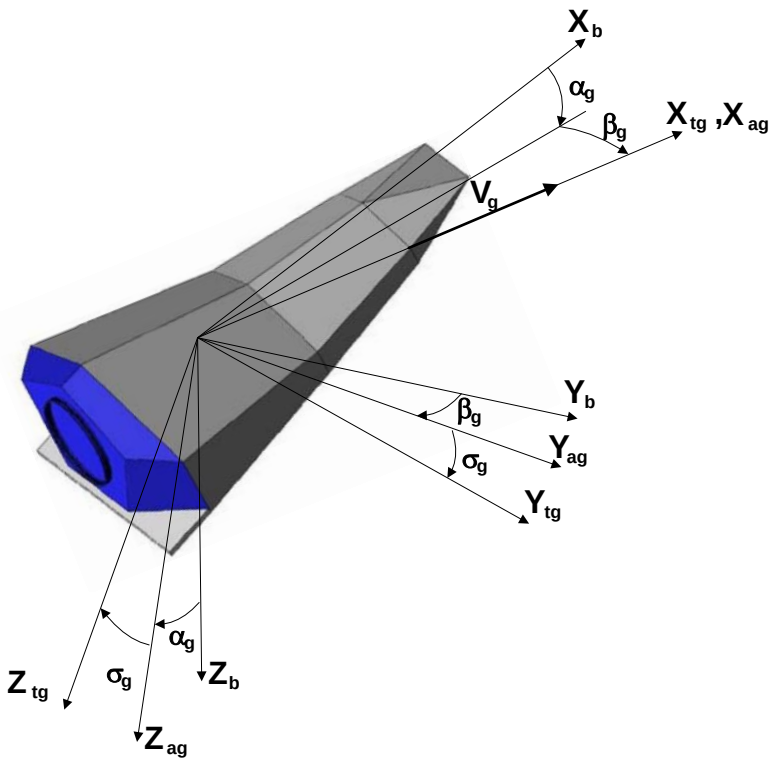


Figure 4.9: SHEFEX-3 Trajectory reference frame.

in the next section will use this representation for the description of the motion.

4.2 Equations of Motion

Once that the reference frames are defined, it is possible to describe the equations of motion. First the full model, containing all the terms, will be described, and its use will be explained. Then, according to different assumptions which can be made, simplified representations of the equations of motion, used over the years for several guidance schemes, will be reported and discussed. The equations of motion derive directly from Newton's second law, and are here expressed in the trajectory reference frame, introduced in Sec. 4.1.8. Since they are used to generate a feasible solution from the perspective of the vehicle's actuation system, additional equations to limit the

controls and the control derivatives are included. When not specified in the title of the sections, the independent variable is time. Moreover it is worth to notice that in Sec. 4.2 no distinction between groundspeed and airspeed variables is made. This difference will be properly addressed in Sec. 4.3, where airspeed and groundspeed variables will be specified, and the relationship between them determined.

4.2.1 Full 3-DOF model in rotating spherical Earth with bank-angle acceleration saturation

The motion of a vehicle in a planetary atmosphere, having limits on the bank angle and its first and second derivative, is described by the following equations:

$$\begin{aligned}
\dot{h} &= V \sin\gamma \\
\dot{\theta} &= \frac{V\cos\gamma\cos\psi}{r\cos\phi} \\
\dot{\phi} &= \frac{V\cos\gamma\sin\psi}{r} \\
\dot{V} &= -D - g\sin\gamma + \omega_{\oplus}^2 r\cos\phi\,(\sin\gamma\cos\phi - \cos\gamma\sin\phi\sin\psi) \\
\dot{\gamma} &= \frac{L\cos\sigma}{V} + \left(\frac{V}{r} - \frac{g}{V}\right)\cos\gamma + 2\omega_{\oplus}\cos\phi\cos\psi + \\
&\quad + \frac{\omega_{\oplus}^2 r}{V}\cos\phi\,(\cos\gamma\cos\phi + \sin\gamma\sin\phi\sin\psi) \\
\dot{\psi} &= \frac{L\sin\sigma}{V\cos\gamma} + \frac{V}{r}\cos\gamma\sin\psi\tan\phi + 2\omega_{\oplus}\,(\tan\gamma\cos\phi\sin\psi - \sin\phi) + \\
&\quad - \frac{\omega_{\oplus}^2 r}{V\cos\gamma}\sin\phi\cos\phi\cos\psi \\
\dot{\sigma} &= u_{\sigma} \\
\dot{u}_{\sigma} &= a_{\sigma}
\end{aligned} \tag{4.3}$$

The states are the altitude h, the longitude θ, the latitude ϕ, the velocity modulus V, the flight-path angle γ, the velocity azimuth angle ψ, the bank angle σ, and the bank-angle rate u_{σ}. L and D are the lift and drag accelerations, given by

$$\begin{aligned}
D &= \frac{\bar{D}}{m} \\
L &= \frac{\bar{L}}{m}.
\end{aligned} \tag{4.4}$$

where $\bar{D}$ and $\bar{L}$ are the drag and lift force, respectively, while m is the mass of the vehicle, which for unpowered vehicle can be assumed to be constant. g is the gravity acceleration, and $\omega_{\oplus}$ is the Earths angular rate, equal to 7.292115 10^{-5} rad/s. For this differential system of equations, the physical input to the system is the bank-angle acceleration a_{σ}. This yields the possibility to easily introduce further limitations in terms of actuators' saturation in the transcription process. What do the last equations

in the systems Eq. (4.3) represent? The answer resides in the way the trajectories are generated. Every optimizer can include upper and lower limits on states and controls, but not on the control derivatives directly. To avoid this problem, the controls are treated as states, and by adding their differential equations, the "role" of control variable is shifted to their rates. With this transformation it is possible to include proper limits on the new controls, i.e., the control rates or control accelerations of the original problem. Therefore, this representation of the controls makes the integration with the flight-control system much smoother. For what regards the angle of attack, if available as control (i.e., if it can be modulated on purpose) t is "hidden" in the aerodynamic accelerations through the aerodynamic coefficients C_L and C_D, and can be used to modulate them.

4.2.2 Full 3-DOF model in rotating spherical Earth with bank-angle rate saturation

The model defined by Eq. (4.3) can be expensive from a computational point of view. Moreover, it may happen that the generated solutions are smooth, and do not require to be forced to satisfy further limits on the bank-angle acceleration, as this is already within the boundaries. In this case, it is possible to remove the last equation of Eq. (4.3), which brings to the following model.

$$
\begin{aligned}
\dot{h} &= V \sin\gamma \\
\dot{\theta} &= \frac{V \cos\gamma \cos\psi}{r \cos\phi} \\
\dot{\phi} &= \frac{V \cos\gamma \sin\psi}{r} \\
\dot{V} &= -D - g \sin\gamma + \omega_{\oplus}^2 r \cos\phi \left(\sin\gamma \cos\phi - \cos\gamma \sin\phi \sin\psi\right) \\
\dot{\gamma} &= \frac{L \cos\sigma}{V} + \left(\frac{V}{r} - \frac{g}{V}\right) \cos\gamma + 2\omega_{\oplus} \cos\phi \cos\psi + \\
&\quad + \frac{\omega_{\oplus}^2 r}{V} \cos\phi \left(\cos\gamma \cos\phi + \sin\gamma \sin\phi \sin\psi\right) \\
\dot{\psi} &= \frac{L \sin\sigma}{V \cos\gamma} + \frac{V}{r} \cos\gamma \sin\psi \tan\phi + 2\omega_{\oplus} \left(\tan\gamma \cos\phi \sin\psi - \sin\phi\right) + \\
&\quad - \frac{\omega_{\oplus}^2 r}{V \cos\gamma} \sin\phi \cos\phi \cos\psi \\
\dot{\sigma} &= u_{\sigma}
\end{aligned}
\tag{4.5}
$$

It is easy to see that Eq. (4.5) have as control the bank-angle rate directly. This simplifies the computation of the trajectories. All the optimal trajectories reportedin this work have been computed by using Eq. (4.5). The control accelerations have been verified, to check the need to use Eq. (4.3). However, for the cases included here this was not necessary, as all the trajectories had control accelerations within the prescribed limits.

4.2.3 Full 3-DOF model in rotating spherical Earth

In a similar fashion to what has been done previously, a further simplification can be made if we remove the differential equation of the bank angle. The result is the model described below:

$$\begin{aligned}
\dot{h} &= V\sin\gamma \\
\dot{\theta} &= \frac{V\cos\gamma\sin\psi}{r\cos\phi} \\
\dot{\phi} &= \frac{V\cos\gamma\cos\psi}{r} \\
\dot{V} &= -D - g\sin\gamma + \omega_{\oplus}^2 r\cos\phi\left(\sin\gamma\cos\phi - \cos\gamma\sin\phi\sin\psi\right) \\
\dot{\gamma} &= \frac{L\cos\sigma}{V} + \left(\frac{V}{r} - \frac{g}{V}\right)\cos\gamma + 2\omega_{\oplus}\cos\phi\cos\psi + \\
&\quad + \frac{\omega_{\oplus}^2 r}{V}\cos\phi\left(\cos\gamma\cos\phi + \sin\gamma\sin\phi\sin\psi\right) \\
\dot{\psi} &= \frac{L\sin\sigma}{V\cos\gamma} + \frac{V}{r}\cos\gamma\sin\psi\tan\phi + 2\omega_{\oplus}\left(\tan\gamma\cos\phi\sin\psi - \sin\phi\right) + \\
&\quad - \frac{\omega_{\oplus}^2 r}{V\cos\gamma}\sin\phi\cos\phi\cos\psi
\end{aligned} \tag{4.6}$$

The model represented by Eq. (4.6) has as control inputs the angle of attack (if it can be modulated), which affects C_L and C_D, and the bank angle directly. This model is less computationally intense than the previous ones. This set of equations has been modified (as discussed in Sec. 4.3) to take the presence of wind into account, and used to integrate the equations of motion during the Monte Carlo campaigns.

4.2.4 3-DOF model in non-rotating spherical Earth

In some cases, it is necessary to simplify the equations of motion to apply particular guidance schemes[7,8]. One of the approximations taken is to neglect the Earth's rotation, having the following set of equations:

$$\begin{aligned}
\dot{h} &= V\sin\gamma \\
\dot{\theta} &= \frac{V\cos\gamma\sin\psi}{r\cos\phi} \\
\dot{\phi} &= \frac{V\cos\gamma\cos\psi}{r} \\
\dot{V} &= -D - g\sin\gamma \\
\dot{\gamma} &= \frac{L\cos\sigma}{V} + \left(\frac{V}{r} - \frac{g}{V}\right)\cos\gamma \\
\dot{\psi} &= \frac{L\sin\sigma}{V\cos\gamma} + \frac{V}{r}\cos\gamma\sin\psi\tan\phi
\end{aligned} \tag{4.7}$$

Once again, the controls are the angle of attack by means of the aerodynamic accelerations L and D, and the bank angle, and this model further simplifies the computational burden required to compute solutions.

4.2.5 Full 3-DOF Model in non-rotating spherical Earth w.r.t. energy

The previous model uses the time as independent variable. A valid alternative is to use the specific energy.

$$E = \left(\frac{V^2}{2} - \frac{\mu_\oplus}{r} \right) \tag{4.8}$$

With this definition, it is possible to transform the independent variable of the set of differential equations. Indeed, if we differentiate Eq. (4.8) w.r.t. time we get

$$\dot{E} = V\dot{V} + \frac{\mu_\oplus}{r^2}\dot{r} \tag{4.9}$$

It is noted that, for a spherical Earth, the identity $\dot{r} = \dot{h}$ holds. Moreover, we can observe that $g = \frac{-\mu_\oplus}{r^2}$. With these identities it is immediate to show that

$$\dot{E} = -DV - gV\sin\gamma + \frac{\mu_\oplus}{r^2}V\sin\gamma = -DV \tag{4.10}$$

Therefore, if we combine Eq. (4.7) with Eq. (4.10), we obtain the following set of differential equations expressed w.r.t. the specific energy E.

$$\begin{aligned}
h' &= -\frac{\sin\gamma}{D} \\
\theta' &= -\frac{\cos\gamma\sin\psi}{Dr} \\
\phi' &= -\frac{\cos\gamma\cos\psi}{Dr} \\
V' &= \frac{1}{V} + \frac{g}{DV}\sin\gamma \\
\gamma' &= \frac{\cos\gamma}{V^2}\left(g - \frac{V^2}{r}\right)\frac{1}{D} - \frac{1}{V^2}\frac{L}{D}\cos\sigma \\
\psi' &= -\frac{L\sin\sigma}{DV^2\cos\gamma} - \frac{1}{Dr}\cos\gamma\sin\psi\tan\phi
\end{aligned} \tag{4.11}$$

The models described by Eqs. (4.7) and (4.11) can be used to compute optimal trajectories with SPARTAN. The corresponding trajectories represent a very good initial guess for solution of the full equations described by (4.3), (4.5), or (4.6).

4.2.6 3-DOF model in non-rotating spherical Earth

For missions where the downrange and the crossrange are more important, it is possible to neglect the differential equations for the longitude and the latitude, obtaining the

following differential equations:

$$\begin{aligned}
\dot{h} &= V \sin\gamma \\
\dot{V} &= -D - g \sin\gamma \\
\dot{\gamma} &= \frac{1}{V} L \cos\sigma + \left(\frac{V}{r} - \frac{g}{V} \right) \cos\gamma \\
\dot{\psi} &= \frac{L \sin\sigma}{V \cos\gamma} + \frac{V}{r} \cos\gamma \cos\psi \tan\phi
\end{aligned} \tag{4.12}$$

This is the reference model used for the development of the Space Shuttle guidance[2]. The input to this system is again the bank angle σ.

4.2.7 3-DOF model in non-rotating spherical Earth w.r.t. energy

Similarly, the system described by Eq. (4.12) can be transformed into an energy-based model, yielding the following model:

$$\begin{aligned}
h' &= -\frac{\sin\gamma}{D} \\
V' &= \frac{1}{V} + \frac{g}{DV} \sin\gamma \\
\gamma' &= \frac{\cos\gamma}{V^2} \left(g - \frac{V^2}{r} \right) \frac{1}{D} - \frac{1}{V^2} \frac{L}{D} \cos\sigma \\
\psi' &= -\frac{L \sin\sigma}{DV^2 \cos\gamma} - \frac{1}{Dr} \cos\gamma \sin\psi \tan\phi
\end{aligned} \tag{4.13}$$

Also in this case the input is the bank angle σ. Why do we need the models represented by Eqs. (4.12) and (4.13)? Because they can be used as models to develop algorithms which combine longitudinal and lateral logic into a complete guidance system. This is the case for instance of the complete Space Shuttle guidance[2], or the algorithm EAGLE[37]. In this work they have been used to integrate the lateral guidance with the longitudinal schemes for the main guidance scheme (Chapter 9), and the back-up guidance scheme (Chapter 10), respectively.

4.2.8 2-DOF model in non-rotating spherical Earth

Finally, the most simplified model which can be implemented is given by three differential equations. They describe the longitudinal motion of a vehicle in terms of h, V, and γ, while the velocity-azimuth angle ψ is neglected.

$$\begin{aligned}
\dot{h} &= V \sin\gamma \\
\dot{V} &= -D - g \sin\gamma \\
\dot{\gamma} &= \frac{1}{V} \left(\frac{V^2}{r} - g \right) \cos\gamma + \frac{D}{V} u
\end{aligned} \tag{4.14}$$

The independent variable in this case is the time. In this model several effects, such as the rotation, or the oblateness of the central body, are neglected. Moreover, this reduced model cannot predict the crossrange, as the differential equation governing the velocity-azimuth angle is excluded. In many cases the input to this model is expressed as vertical component of lift acceleration scaled by the drag acceleration. Mathematically, it is described as

$$u = \frac{C_L}{C_D} \cos \sigma \tag{4.15}$$

Equation (4.15) shows the duality of the control for entry-guidance schemes. Depending from the methods, one can choose to express the control as u, and then extract the bank angle from the inversion of Eq. (4.15) (this is the case of the back-up guidance developed in Chapter 10, or determine directly the bank angle (as done for the main guidance described in Chapters 8 and 9). Another observation one can make is that this model does not include any information about ψ as this variable does not affect the results for the longitudinal dynamics, and its differential equation is here not included. This agrees with the limit for this model to predict the crossrange. Its main utility is the capability to predict instead the constraints, like the dynamic pressure, and the heat rate, which are always described with the sole use of the longitudinal states. Moreover, this model will be the starting point for the design of the feedback controller, based on sliding mode control theory, presented in Chapter 9.

4.2.9 2-DOF model in non-rotating spherical Earth w.r.t. energy

As done for the previous models, also this system can be converted in an energy-domain based system, yielding:

$$\begin{aligned} h' &= -\frac{\sin \gamma}{D} \\ V' &= \frac{1}{V} + \frac{g}{DV} \sin \gamma \\ \gamma' &= \frac{\cos \gamma}{V^2} \left(g - \frac{V^2}{r} \right) \frac{1}{D} - \frac{1}{V^2} u \end{aligned} \tag{4.16}$$

This model will be used in Chapter 10 for the development of the longitudinal back-up guidance. As before, (Eq. (4.14)), also this model does not account for the crossrange. Since all the models used in the work have been defined, it is possible to introduce the wind in the equations of motion. This is the focus of the next section.

4.3 Integration of Wind in the Equations of motion

This section illustrates how the wind has been included in the equations of motion. The scheme shown in Fig. 4.10 shows the implemented scheme, similar to what has been proposed by Mooij[36], with a slight modification which takes the Horizontal Wind Model 07 interface into account.
In the following sections we will describe all the submodules of the scheme of Fig. 4.10.

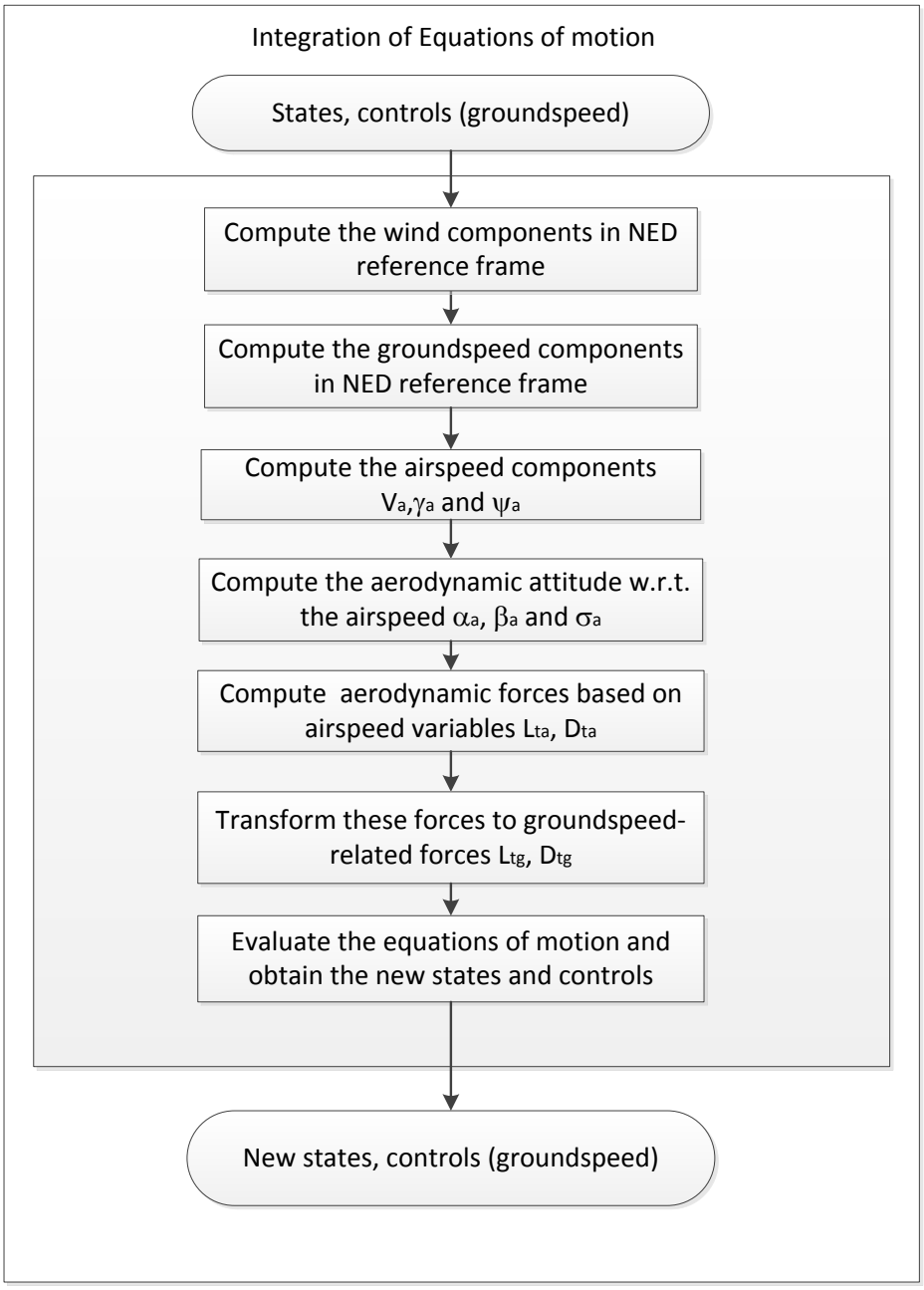


Figure 4.10: Integration of equations of motion in presence of wind.

4.3.1 Computation of wind components in NED reference frame

The first step is the computation of the wind parameters, meant as velocity components V_w, γ_w, and ψ_w. The wind components are provided as meridional and zonal components.

Their orientation is shown in Fig. 4.11. We can convert the wind components from

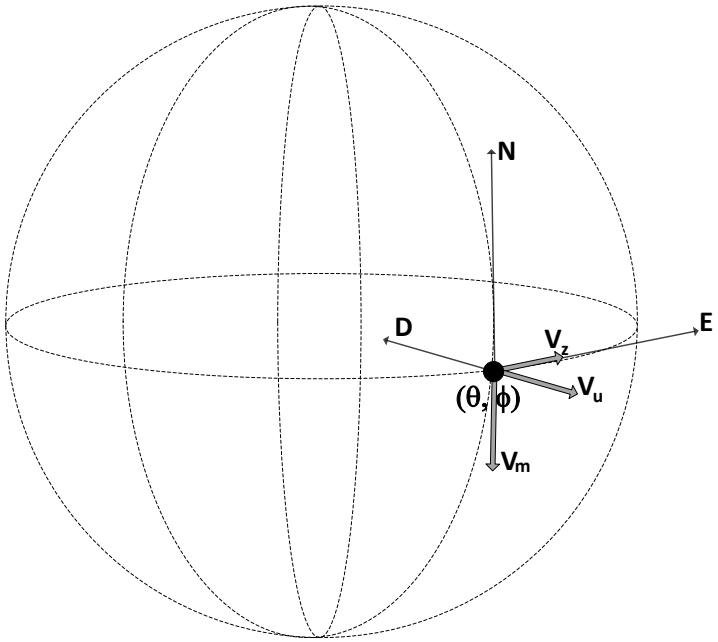


Figure 4.11: Meridional and zonal components of wind.

meridional and zonal components into a North-East-Down representation. It is easy to see that the transformation is given by

$$\mathbf{V}_w^{ned} = \mathbf{T}_{mzu}^{ned} \mathbf{V}_w^{mzu} \tag{4.17}$$

where the rotation matrix T_{mzu}^{ned} is

$$\mathbf{T}_{mzu}^{ned} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4.18}$$

The vector $\mathbf{V}_w^{mzu}$ is

$$\mathbf{V}_w^{mzu} = \begin{bmatrix} V_m \\ V_z \\ 0 \end{bmatrix}^T \tag{4.19}$$

4.3.2 Computation of groundspeed components in NED reference frame

Let us transform the groundspeed-based components of velocity from spherical coordinates into North-East-Down representation too. We can write

$$\mathbf{V}_g^{ned} = \begin{bmatrix} V_{N,g} \\ V_{E,g} \\ V_{D,g} \end{bmatrix} = \begin{bmatrix} V\cos\gamma_g\cos\psi_g \\ V\cos\gamma_g\sin\psi_g \\ -V\sin\gamma_g \end{bmatrix} \tag{4.20}$$

Now both the groundspeed vector $\mathbf{V}_g$ and the wind velocity vector $\mathbf{V}_w$ are expressed in the same reference frame. They can be combined to obtain the airspeed vector $\mathbf{V}_a$.

4.3.3 Computation of airspeed components

Since the wind vector and the groundspeed vector are expressed in the same reference frame the airspeed vector can be simply obtained as

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w \tag{4.21}$$

The airspeed vector is essential for a correct computation of the aerodynamic forces, which depend on the effective relative speed w.r.t. the airflow. We can therefore compute the spherical coordinates of velocity w.r.t. the air. To do it, we can use the inverse of Eq. (4.20), referred this time to the airspeed V_a.

$$\begin{aligned} V_a &= \sqrt{V_{a,N}^2 + V_{a,E}^2 + V_{a,D}^2} \\ \gamma_a &= -\sin^{-1}\left(\frac{V_{a,D}}{V_a}\right) \\ \psi_a &= -\tan^{-1}\left(\frac{V_{a,E}}{V_{a,N}}\right) \end{aligned} \tag{4.22}$$

4.3.4 Computation of the aerodynamic attitude w.r.t. the airspeed

To complete the correct computation of the aerodynamic forces, we need to compute the aerodynamic attitude w.r.t. the airspeed. In other words, we are interested to compute the airspeed-based angle of attack α_a, the airspeed-based σ_a, and theoretically the side-slip angle β_a. However, it is assumed that both the groundspeed side-slip angle β_g and the airspeed side-slip angle β_a are zero, as this will be a main requirement for the attitude control system. Consistently with what has been done by Mooij[36], we can compute these components as follows:

The transformation matrix from the vertical reference frame to the body reference frame is expressed as

$$\mathbf{T}_{lvlh}^{b} = \mathbf{T}_2(\alpha_g)\,\mathbf{T}_3(-\beta_g)\,\mathbf{T}_1(-\sigma_g)\,\mathbf{T}_2(\gamma_g)\,\mathbf{T}_3(\psi_g) \tag{4.23}$$

The same matrix can be decomposed as

$$\mathbf{T}_{lvlh}^{b} = \mathbf{T}_{ta}^{b}\,\mathbf{T}_{lvlh}^{ta} \tag{4.24}$$

where the matrices $\mathbf{T}_{lvlh}^{ta}$ and $\mathbf{T}_{ta}^{b}$ represent the transformations from the LVLH to the airspeed-based trajectory reference frame, and from the trajectory to the body reference frame, respectively.

Consistently with Eq. (4.23), the matrix $\mathbf{T}_{ta}^{b}$ can be expressed as

$$\mathbf{T}_{ta}^{b} = \mathbf{T}_2(\alpha_a)\, \mathbf{T}_3(-\beta_a)\, \mathbf{T}_1(-\sigma_a) \tag{4.25}$$

which contains the variables we are interested to. Written out, the matrix $\mathbf{T}_{ta}^{b}$ is

$$\mathbf{T}_{ta}^{b} = \begin{bmatrix} c_{\alpha_a} c_{\beta_a} & -c_{\alpha_a} s_{\beta_a} c_{\sigma_a} & c_{\alpha_a} s_{\beta_a} s_{\sigma_a} - s_{\alpha_a} c_{\sigma_a} \\ s_{\beta_a} & c_{\beta_a} c_{\sigma_a} & -c_{\beta_a} s_{\sigma_a} \\ s_{\alpha_a} c_{\beta_a} & c_{\alpha_a} s_{\sigma_a} - s_{\alpha_a} s_{\beta_a} c_{\sigma_a} & c_{\alpha_a} c_{\sigma_a} + s_{\alpha_a} s_{\beta_a} s_{\sigma_a} \end{bmatrix} \tag{4.26}$$

where c and s stand for $\cos()$ and $\sin()$, respectively.We can compute this rotation matrix, and then extract the angles α_a, β_a, and σ_a. The matrices from airspeed-based trajectory reference frame to LVLH, and from LVLH to body reference frame are

$$\mathbf{T}_{ta}^{lvlh} = \mathbf{T}_3(-\psi_a)\, \mathbf{T}_2(-\gamma_a) \tag{4.27}$$

$$\mathbf{T}_{lvlh}^{b} = \mathbf{T}_2(\alpha_g)\, \mathbf{T}_3(-\beta_g)\, \mathbf{T}_1(-\sigma_g)\, \mathbf{T}_2(\gamma_g)\, \mathbf{T}_3(\psi_g) \tag{4.28}$$

So, we can compute $\mathbf{T}_{ta}^{b}$ as

$$\mathbf{T}_{ta}^{b} = \mathbf{T}_{lvlh}^{b}\, \mathbf{T}_{ta}^{lvlh} \tag{4.29}$$

We can relate the matrix in Eq. (4.26), with the one computed according to Eq. (4.29) to extract the airspeed, attitude angles:

$$\begin{aligned} \alpha_a &= \tan^{-1}\left(\frac{\mathbf{T}_{ta}^{b}(3,1)}{\mathbf{T}_{ta}^{b}(1,1)}\right) \\ \sigma_a &= \tan^{-1}\left(-\frac{\mathbf{T}_{ta}^{b}(2,3)}{\mathbf{T}_{ta}^{b}(2,2)}\right) \end{aligned} \tag{4.30}$$

4.3.5 Computation of aerodynamic forces based on airspeed variables

Given the effective airspeed-related variables, the effective aerodynamic forces generated by the vehicle during the atmospheric entry are

$$\begin{aligned} L_a &= \frac{1}{2}\rho V_a^2 \frac{S}{m} C_{L,a} \\ D_a &= \frac{1}{2}\rho V_a^2 \frac{S}{m} C_{D,a} \end{aligned} \tag{4.31}$$

where $C_{L,a}$ and $C_{D,a}$ are the aerodynamic coefficients computed with airspeed variables.

4.3.6 Transformation of aerodynamic forces into groundspeed-based aerodynamic forces

The aerodynamic forces computed in the previous section are defined w.r.t. the actual airspeed orientation. However, in the equations of motion these forces are expressed w.r.t. the groundspeed-based trajectory reference frame. With the airspeed-based aerodynamic force vector defined as $\mathbf{F}_{ta}$, and the groundspeed-based aerodynamic force vector $\mathbf{F}_{tg}$, we can write

$$\mathbf{F}_{tg} = \mathbf{T}_{ta}^{tg}\, \mathbf{F}_{ta} \tag{4.32}$$

where the matrix $\mathbf{T}_{ta}^{tg}$ transforms the vector expressed in the airspeed-based trajectory reference frame into the groundspeed-based trajectory reference frame, consistent with the integration of the equations of the motion[36].

$$\mathbf{T}_{ta}^{tg} = \mathbf{T}_1(-\sigma_g)\, \mathbf{T}_2(\gamma_g)\, \mathbf{T}_3(\psi_g - \psi_a)\, \mathbf{T}_2(-\gamma_a)\, \mathbf{T}_1(\sigma_a) \tag{4.33}$$

4.3.7 Evaluation of equations of motion

The lift and drag forces can be used for the evaluation of the equations of the motion. It is worth to stress that, given its semistochastic nature, and without planning proper measurements, the wind is included for the verification of the behavior of the guidance system, and not for its design. In particular, for the results reported in the Monte Carlo campaigns described in Chapter 11, the differential system of Eq. (4.6) has been used.

Chapter 5

Entry Guidance: State of the Art

The objective of this chapter is to provide an overview of the evolution of the main techniques for the Entry guidance problem developed so far. The idea is to start with the first closed-form approaches developed for the Apollo missions[1] and further for the Space Shuttle program[2], to arrive at the most recent and valid solutions developed in the last decades. It is worth to stress that, given the vastness of the topic, only a limited, but relevant part of the literature can be briefly reported here. In Sec. 5.1 the Space Shuttle entry guidance is briefly described. The evolution of this method led to the drag-energy method, summarized in Secs. 5.2-5.4. Different approaches, based on predictor-corrector algorithms, and constraint-tracking laws, have been proposed over the years, and are the focus of Secs. 5.5 and 5.6, respectively. The use of optimal control in the frame of entry guidance is described in Sec. 5.7. Finally, further methods, which can be used are briefly mentioned in Sec. 5.8.

5.1 Closed-Form Approach: the Space Shuttle Guidance

The Space Transportation System, also known as the Space Shuttle, has been one of the greatest successes of NASA. Given the variability of the missions performed over the years, its entry guidance system was conceived to provide a good trade-off between the reliability and the flexibility required for different entry scenarios. Moreover, since it was a crewed vehicle, special attention was taken to prevent strong load factors, limited to 2.5 g. The Space Shuttle guidance, as many other approaches, is separated in two sub-modules, the longitudinal and lateral guidance.

5.1.1 Longitudinal Guidance

For the entire entry phase, the angle of attack is scheduled to follow a reference profile, shown in Fig. 5.1. For the development of the Space Shuttle entry guidance, an initial angle of attack of about 40 deg was selected, while the operational angle of attack profile was slightly smaller. We can see that the angle of attack remains at high values for the first phase of the entry, and then it is reduced. The reason for this choice, common to

different entry guidance schemes, is the need to protect the vehicle from the elevated value of the heat flux. In this way, the thermal-protection shield can correctly work. To realize the prescribed range needed to reach the runway, Harpold and Graves[2]

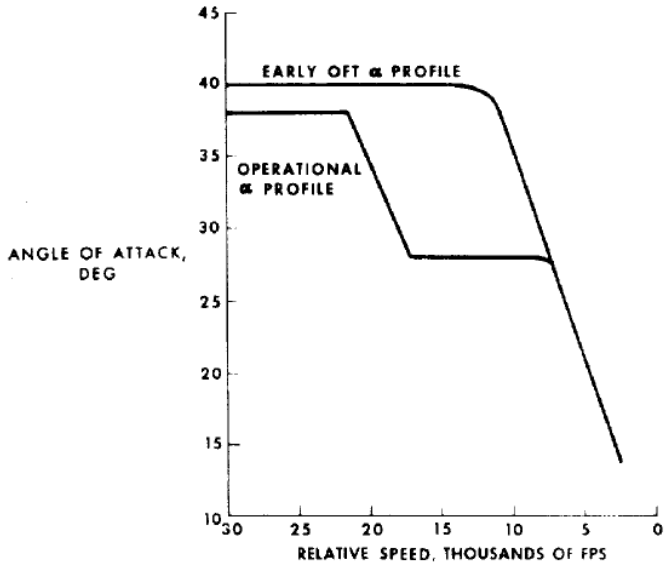


Figure 5.1: Nominal angle of attack profile for the Space Shuttle entry[2].

conceived the range as function of a reference drag acceleration profile to be tracked. This is a significant evolution with respect to the Apollo guidance[1], where after the initial roll needed to ensure capture (with the capsule tilted down), a drag-constant segment was tracked. In the Space Shuttle guidance, however, there are five different segments, which form the reference drag to be tracked. These segments are defined in the drag-velocity plane, and are:

- First Quadratic;
- Second Quadratic;
- Pseudo-Equilibrium Glide;
- Constant Drag;
- Drag-Energy linear;

A typical profile is depicted in Fig. 5.2, where all the constraints acting on the system have been conveniently represented in the drag-velocity space. Specifically, it is possible to determine the upper drag limit according to the heat flux, the load factor and the dynamic pressure, while the lower limit is determined by the need to have sufficient control authority, which turns into a minimum value for the lift able to sustain the

vehicle's weight; since the angle of attack follows a predefined profile, and the C_L/C_D ratio is function of α, the minimum lift turns into a minimum value of drag.

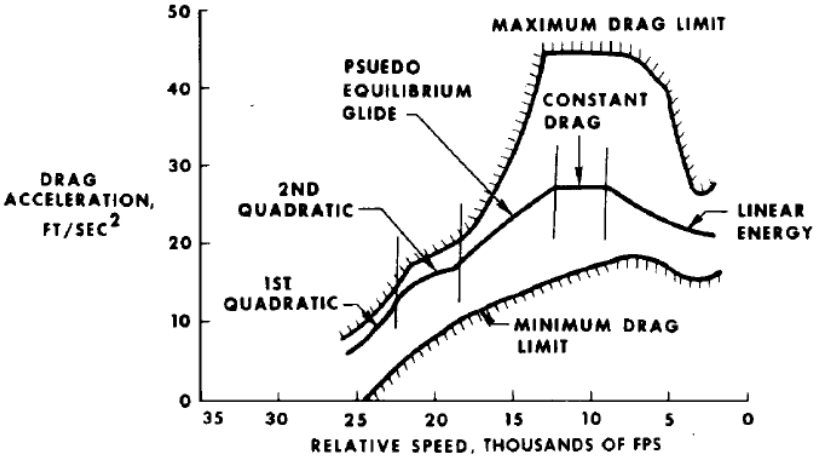


Figure 5.2: Typical drag-velocity diagram for the Space Shuttle entry [2].

It is then possible to compute the feasible space where to define the nominal drag profile with no risk to violate the constraints acting on the system. The initial value of the drag profile is completely determined by the longitudinal state of the vehicle at the entry interface. The profile can then be adapted onboard to null possible range errors by adjusting the parameters defining the various drag segments flown during the entry, as it is shown in Fig. 5.3. From Fig. 5.3 one can observe that, for each phase, an increase of the drag acceleration will reduce the total range, while a decrease will yield an increase in the range capabilities of the vehicle. The drag acceleration and its derivatives can be linked to the altitude, the velocity and the flight-path angle. Therefore, for each phase, the current segment of the drag-velocity profile can be raised or lowered if a shorter or a longer range is required, respectively. Since the drag as function of velocity is described using simple analytical functions, it is possible to determine the range with analytical or numerical relationships. For instance, for the phase having constant drag (also in this case no wind is included, therefore, no distinction between groundspeed- and airspeed-based variables is made in this context) we have

$$D = D_{const} \tag{5.1}$$

and its range contribution will be

$$R = \frac{V^2 - V_F^2}{D_{const}} \tag{5.2}$$

where V is the current velocity, and V_F is the velocity at the end of the specific entry segment. Once the drag profile is set, it is possible to derive the altitude rate $\dot{h}$ and the

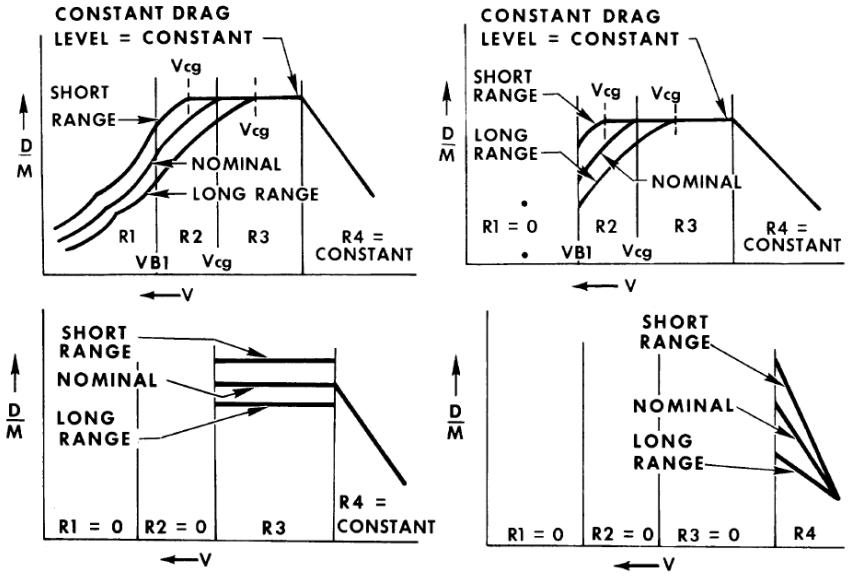


Figure 5.3: Space Shuttle drag profile adaptation[2].

L/D ratio as[2]

$$\dot{h} = -h_s \left(\frac{\dot{D}}{D} + \frac{2D}{v} - \frac{\dot{C}_D}{C_D} \right) \tag{5.3}$$

$$\begin{aligned} \ddot{D} \;=\; & \dot{D}\left(\tfrac{\dot{D}}{D} + \tfrac{3D}{V}\right) - \tfrac{4D^3}{V^2} - \tfrac{D}{h_s}\left(\tfrac{V^2}{r} - g\right) \\ & -\tfrac{D^2}{h_s} L/D - D\tfrac{\dot{C}_D}{C_D}\left(\tfrac{\dot{C}_D}{C_D} - \tfrac{D}{V}\right) + D\tfrac{\ddot{C}_D}{C_D} \end{aligned} \tag{5.4}$$

where:

- h_s is the scale factor used to model the density as function of the altitude;
- $\dot{()}$ and $\ddot{()}$ are the first and second time derivatives;
- L and D are the lift and drag accelerations;
- r is the radial position;
- g is the gravity acceleration;
- C_D is the drag coefficient;

and the unknown terms are $\dot{h}$ and L/D. The derived model will not perfectly match the inflight conditions. Uncertainties, disturbances, inaccuracies and delays will generate a difference w.r.t. the reference drag profile, and must be properly compensated for. The

tracking law implemented for the Space Shuttle is a non-conventional PID controller that computes the commanded $(L/D)_c$ ratio as

$$(L/D)_c = (L/D) + f_1 (D - D_0) + f_2 \left(\dot{h} - \dot{h}_0\right) + f_4 \int_0^{t_F} (D - D_0)\, d\tau \tag{5.5}$$

The commanded angle of attack and bank angle can be c

$$\alpha_c = \alpha_0 + C_D \frac{(D_0 - D)}{f_{10}} \tag{5.6}$$

$$\sigma_c = cos^{-1} \left[\frac{(L/D)_c}{L/D}\right] + f_{11} (\alpha_c - \alpha_0) \tag{5.7}$$

where σ_c is the commanded bank angle, $(L/D)_c$ and L/D are the commanded and estimated lift-to-drag ratios, α_c and α_0 are the commanded and reference angle of attack, D_0 and D are the reference and estimated drag accelerations, and the f_i terms are gains selected according to the mission profile and the vehicle characteristics. The last term in Eq. (5.7) is included to drive the modulated angle of attack back to the reference α, on a long term basis[2]. With the use of the Eqs. (5.3)-(5.7), the longitudinal guidance is completely determined.

5.1.2 Lateral Guidance

The lateral guidance of the Space Shuttle is based on the idea to define an admissible bandwidth in terms of azimuth errors. The deadband defined for the Space Shuttle is depicted in Fig. 5.4, where the upper and lower limits represent the maximum errors tolerated. If the azimuth error exceeds one of the limits, a bank reversal (a rapid change of sign of the bank angle) is commanded. This is possible because the longitudinal dynamics is dominated by the cosine of the angle, while the azimuth error depends on the sign of its sine. Therefore, if the change of sign is performed, the lateral error decreases, with a reduced or no impact on the longitudinal performance of the guidance system.

The azimuth error is defined as the angle between the plane formed by the position and the velocity vectors of the spacecraft and the vector obtained connecting the spacecraft to the target point on a heading alignment circle placed close to the runway, as shown in Fig. 5.5, where one can see that, if the azimuth error is kept to small values, the vehicle will fly towards the heading alignment circle.

The Space Shuttle Entry guidance has demonstrated to work very well, and has become the standard in entry guidance. Moreover, it has provided the basis for researchers to further explore this concept, as we will see in the next section.

5.2 Drag vs Energy Approach

In Sec. 5.1.1 it was briefly mentioned that, for the Space Shuttle drag reference profile, the last phase was scheduled as a linear function of the specific energy, that is

$$D = D_i + D_E\,(E - E_i) \tag{5.8}$$

where D_i, E_i and D_E are constant values that allow for defining linear relationships between energy and drag. This choice is justified for accuracy reasons: when the range

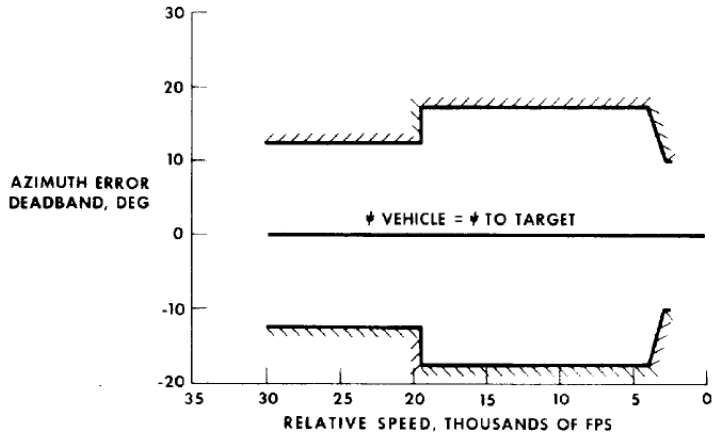


Figure 5.4: Space Shuttle azimuth deadband[2].

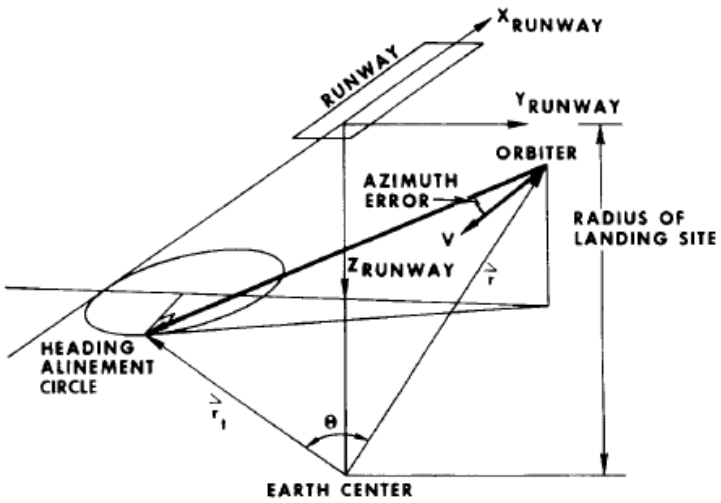


Figure 5.5: Entry azimuth error geomentry[2].

is predicted in the drag-energy space, the accuracy of the prediction is larger than when done in the drag-velocity space. When the velocity is taken as independent variable, the range prediction associated to a given drag profile is

$$R = \int_{V_i}^{V_f} \frac{v \cos\gamma}{-D - g \sin\gamma} dv \cong - \int_{V_i}^{V_f} \frac{v}{D} dv \tag{5.9}$$

while when the energy as independent variable is selected, we have

$$R = \int_{E_i}^{E_f} -\frac{\cos\gamma}{D} dE \cong \int_{E_i}^{E_f} -\frac{1}{D} dE \tag{5.10}$$

From the comparison of Eqs. (5.9) and (5.10) it is possible to see that the range predicted in the energy domain gives more accurate results, as the term $g \sin\gamma$ is included directly in E, and is not neglected as in Eq. (5.9). Other advantages are the consistency of the drag profile, now entirely described in terms of energy, and this description of the motion gives already a natural interface for the TAEM[38]. This leads to the possibility to describe the entire drag profile, and not only the last segment in the drag-energy domain. Mease et al.[7,39] and Saraf et al.[37] developed an entry guidance scheme, which is entirely represented in the drag-energy domain. The extended mathematical formulation of the equations of motion and the constraints in the drag-energy domain can be found in Chapter 4. For this specific method, a drag reference profile is used for the planned trajectory. The choice for this profile can be done either using linear pieces linked together, or solving an optimal-control problem that takes all the constraints acting on our system into account.

For simplifying the formulation, some aerodynamic coefficients can be considered constant or linear functions of the energy. This gives the possibility to neglect the first and/or the second derivative of the drag coefficient, respectively, which is usually a standard assumption for the study of the atmospheric entry of vehicles like the Apollo capsules[7,37]. High/medium lifting vehicles have usually more complex geometry, and, consequently, more complicated aerodynamic properties. In the formulation of the entry guidance scheme, Saraf et al.[37] divide the problem into two sub-problems, that, linked together, can be solved with an iterative process: the trajectory length sub-problem, and the curvature sub-problem. Fig. 5.6 illustrates the trajectory length and curvature sub-problems, which deal with the problem to generate a trajectory of a given range, and which has the heading angle within some prescribed limits.

5.2.1 Determination of the trajectory length and the longitudinal control law

The objective of the trajectory length sub-problem is to find a reference drag profile that satisfies the requirements on the range, and to extract the controls associated with it.

$$R = \int_{E_i}^{E_f} -\frac{1}{D} dE = R_{ref} \tag{5.11}$$

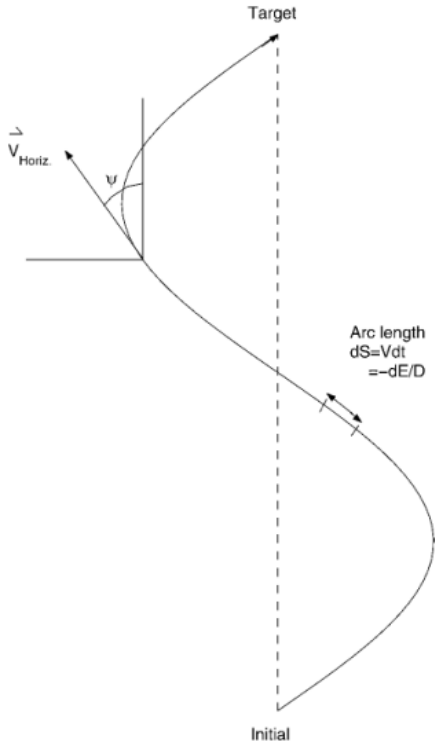


Figure 5.6: Trajectory Length and Curvature sub-problems[39].

where R is the current range, and R_{ref} is the nominal range to be covered.

Figure 5.6 shows an example of the two sub-problems, implemented for the X-33 entry vehicle. The range is represented by the distance between the initial point and the target. However, every modulation of the bank angle will have an impact on the heading angle, and will modify the curvature of the trajectory. The logic to shape the curvature of the trajectory will be the focus of the curvature sub-problem, described in Sec. 5.2.2. However, for the trajectory length sub-problem, the bank angle is needed. It is possible to demonstrate that it can be related to the second derivative of the drag acceleration with respect to the energy. Specifically, we can write[7,37]

$$D'' = a_D + b_D(L/D)\cos\sigma \tag{5.12}$$

where a and b can be computed as

$$\begin{aligned} a_D &= D\left(\frac{C_D''}{C_D} - \frac{{C_D'}^2}{C_D^2}\right) + D'\left(\frac{C_D'}{C_D} + \frac{2}{V^2}\right) \\ &\quad -\frac{4D}{V^4} + \frac{1}{DV^2}\left(\frac{1}{h_s} - \frac{2g}{V^2}\right)\left(g - \frac{V^2}{r}\right) \\ b_D &= -\frac{1}{V^2}\left(\frac{1}{h_s} + \frac{2g}{V^2}\right) \end{aligned} \tag{5.13}$$

where D and D' are the drag acceleration and its derivative w.r.t. the energy, C_D, C'_D and C''_D are the drag coefficient and its derivatives w.r.t. the energy, and h_s is the scale height. An example of drag-energy profile is plotted in Fig. 5.7. The energy is represented in non-dimensional form (a proper definition will be given in Chapter 10), and we can see that the planned drag profile is defined in the feasible space, i.e., between the minimum and maximum drag boundaries.

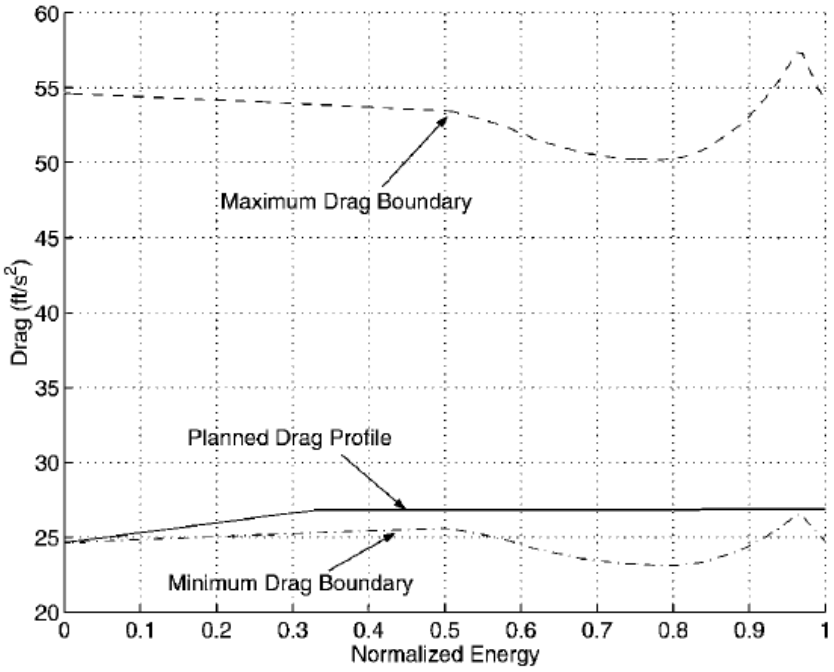


Figure 5.7: Drag Energy piecewise linear profile for the X-33[7].

Alternatively, it is possible to extract the control from the condition of equilibrium glide, defined as the control which satisfies the conditions of small flight-path angle and flight-path angle rate.

$$\gamma,\ \dot{\gamma} \cong 0 \Rightarrow \frac{L}{D}\cos\sigma - \frac{1}{D}\left(g - \frac{v^2}{r}\right) \cong 0 \tag{5.14}$$

Once the controls are extracted either by Eq. (5.12), or by Eq. (5.14), the range can be computed assuming the great-circle approximation, which consists of approximating the radial position of the spacecraft (usually in non-dimensional form) as a constant value associated with the final interface altitude[2,39].

$$r_{gca} = r_F \tag{5.15}$$

This assumption is used only for the computation of the second derivative of drag. However, for each drag acceleration and energy level, and assuming that the angle of

attack is known, it is possible to extract a corresponding pair r, v. The flight-path angle can be related to the first derivative of drag acceleration instead. The first derivative of drag acceleration w.r.t. the energy is defined as

$$D' = D\left(\frac{\rho'}{\rho} + 2\frac{V\prime}{V} + \frac{C_D'}{C_D}\right) \tag{5.16}$$

If we combine Eq. (5.16) with the system of defined by Eq. (4.13), and taking the exponential model

$$\rho = \rho_0 e^{-h/h_s} \tag{5.17}$$

into account, it is possible to extract the flight-path angle associated with the defined drag profile.

$$\sin\gamma = \frac{D' - D\left(\frac{C_D'}{C_D} + \frac{2}{v^2}\right)}{\left(\frac{1}{h_s} + \frac{2g}{v^2}\right)} \tag{5.18}$$

All the longitudinal variables can then be computed, and the trajectory length sub-problem is solved.

5.2.2 Determination of the trajectory curvature

The following step is to solve the curvature sub-problem, which consists of the determination of the sign of the bank angle, which minimizes the error in terms of desired final heading angle ψ. Given a desired value for the flight-path heading angle ψ_F, the reduced set of differential Eqs. (5.19)-(5.21) can be integrated to determine where to perform the bank reversal which realizes the condition $\psi(t_F) = \psi_F$.

$$\theta' = -\frac{1}{D}\frac{\cos\gamma\cos\psi}{\hat{r}\cos\phi} \tag{5.19}$$

$$\phi' = -\frac{1}{D}\frac{\sin\psi}{\hat{r}} \tag{5.20}$$

$$\psi' = \frac{1}{v^2}\frac{L\sin\psi}{D} + \frac{1}{D}\frac{\cos\psi\tan\phi}{\hat{r}} \tag{5.21}$$

The solution to the curvature sub-problem will modify the trajectory length, and viceversa. Then, an iterative process for obtaining a specific tolerance on the trajectory length and the final desired azimuth angle can be set.

5.3 Determination of optimal trajectory

Mease[39] and Lu[8,40,41] proposed alternative formulations based not on the iterative process proposed in the previous two sections, but based on the design of drag profiles, which satisfy a given cost function, for instance, the heat load, which can be expressed in time domain as

$$J = \int_{t_0}^{t_f} \sqrt{(}\rho) V^3 d\tau \tag{5.22}$$

or, in the energy domain, as[41]

$$J = \int_{E_0}^{E_f} \frac{V}{\sqrt{(D)}} dE \cong \sqrt{2} \int_{E_0}^{E_f} \sqrt{\frac{1/\hat{r} - e}{D}} dE \tag{5.23}$$

The advantage of this approach is to generate a drag-profile which not only satisfies the requirements, but is also optimal w.r.t. a given criterion.

5.4 Drag-Energy Tracking Controller

The previous sections provide a reference solution. However, having a feasible trajectory and the controls that realize it is not enough. Inflight conditions can be significantly different from the nominal ones. Moreover, models cannot exactly predict variables like density or aerodynamic coefficients, and some of them are difficult to measure with sensors. Further uncertainties will act on the real system, having as a result a scenario that could be far away from the scheduled one. For this reason, once the open-loop control has been computed, it is necessary to develop a tracking controller. Over the years, several interesting techniques were proposed. Saraf et. al.[37] proposed a modified version of the tracking controller developed for the Shuttle.

$$u_D = \frac{1}{b_D} \left(D''_{ref} - a_D + 2\zeta\omega(D'_{ref} - D') + \omega^2(D_{ref} - D) + k_1 \int_{E_0}^{E_F} (D_{ref} - D) dE \right) \tag{5.24}$$

Eq. (5.24) represents a combined feedforward-feedback law. The term

$$\frac{1}{b_D} \left(D''_{ref} - a_D \right) \tag{5.25}$$

represents the control which satisfied the drag curvature Eq. (5.12). This means that, in nominal conditions, the control computed through Eq. (5.25) will theoretically generate a drag profile equal to the nominal one, which satisfies all the requirements. The second part of Eq. (5.24), that is

$$\frac{1}{b_D} \left(2\zeta_D\omega_D(D'_{ref} - D') + \omega_D^2(D_{ref} - D) + k_{D,i} \int_{E_0}^{E_F} (D_{ref} - D) dE \right) \tag{5.26}$$

together with Eq. (5.25) defines a second-order system having damping ratio ζ_D and natural frequency ω_D. The control computed through Eq. (5.24) imposes a specific error dynamics to the drag, which can be shaped according to the parameters ζ_D and ω_D. Moreover, similarly to the Space Shuttle guidance, the integral term, proportional to $k_{D,i}$, is added to improve the steady-state error. This is not the unique choice. The tracking controller can be formulated directly in terms of states, as proposed by Roenneke[38,42], which designed a traditional tracking approach of the longitudinal state vector, which error is

$$\delta\mathbf{x} = \begin{bmatrix} h - h_{ref} \\ v - v_{ref} \\ \gamma - \gamma_{ref} \end{bmatrix} \tag{5.27}$$

More specifically, Roenneke[38] proposed to track the reference states with the use of a gain-scheduled linear quadratic regulator. This means that the nonlinear dynamics is linearized around the nominal trajectory, so that N LTI systems can be derived. Each LTI system at time t_k can be represented as a set of matrices $\mathbf{A}_k$,$\mathbf{B}_k$,$\mathbf{C}_k$,$\mathbf{D}_k$.

$$f(t_k, \mathbf{x}_k, \mathbf{u}_k) \longrightarrow [\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k, \mathbf{D}_k] \,,\ k = 1, \cdots, N \tag{5.28}$$

Then, for each of this system, an LQR controller is synthesized. The resulting gains are properly interpolated, taking as independent variable for instance, the energy or the dynamic pressure. A drawback of this method is related to the stability of the nonlinear system, which can be proved in the specific nodes where the controller is synthesized, but not between the nodes. For this reason, usually, extensive Monte Carlo campaigns are performed to prove that the nonlinear controller is stable and robust under a certain number of disturbances and uncertainties acting on the system.

5.5 Predictor-Corrector Approach

Lu[9],[40] proposed different applications of the predictor-corrector algorithm for the reentry of non-powered vehicles. As the name suggests, the method is based on a two-step process. In all the cases we are interested in a specific final state, represented by

$$h(t_F) = h_F \tag{5.29}$$

$$V(t_F) = V_F \tag{5.30}$$

$$R(t_F) = R_F \tag{5.31}$$

where the altitude h, the velocity modulus V, and the range R are evaluated at the final time t_F. The module of the bank angle is parameterized as a linear function of the energy,

$$|\sigma| = \sigma_0 + \frac{E - E_0}{E_F - E_0}(\sigma_F - \sigma_0) \tag{5.32}$$

where σ_0 is a parameter to be found, and σ_F is a constant. At each energy state E_0, the predictor algorithm numerically integrates the equations of motion such that the conditions (5.29)-(5.31) at a given final energy (that is, radius and velocity), are realized. Specifically, the problem can be treated either as a zero-finding problem[43],

$$z(\sigma_0) = R(E_F) - R_F = 0 \tag{5.33}$$

or as a minimization problem[44], where the function to be minimized is

$$f(\sigma_0) = \frac{1}{2} z(\sigma_0)^2 = \frac{1}{2}[R(E_F - R_F]^2 \tag{5.34}$$

The conditions (5.29), and (5.30) are included in the process as they define the final value of the energy

$$E_F = \frac{\mu_\oplus}{r_F} - \frac{V_F^2}{2} \tag{5.35}$$

After the prediction step, a residual in the solution of Eq. (5.33), or alternatively, of Eq. (5.34) will occur. The corrector step consists then of updating the parameter σ_0 by a step-controlled Gauss-Newton method,

$$\sigma_0^{k+1} = \sigma_0^k - \lambda_k \frac{\partial f(\sigma_0^k)/\partial \sigma_0}{\left[\partial z(\sigma_0^k)/\partial \sigma_0\right]^2} \tag{5.36}$$

where λ_k is the step, and is chosen equal to $1/2^i$, where i is the smallest integer so that

$$f(\sigma^{k+1}) \leq f(\sigma^k) \tag{5.37}$$

Eq. (5.36) uses the step to ensure the quadratic convergence of the nonlinear function $f(\sigma)$ to 0. Once σ_0 is found, the sign of the bank angle is determined using the bank-reversal logic already seen in the Space Shuttle entry guidance. The method has been enhanced over the years[9,44] with the integration of the quasi-equilibrium glide condition. If we refer to L and $g = 1/r^2$ as non-dimensional lift and gravity accelerations, respectively, the non-dimensional equation representing the equilibrium-glide condition becomes

$$\dot{\gamma} = L\cos\sigma + \left(v^2 - \frac{1}{r}\right)\frac{\cos\gamma}{r} = 0 \tag{5.38}$$

which is valid if the Earth's rotation is neglected. In this case, it is possible to write the flight-path angle and the altitude rate as

$$\gamma_{QEGC} = \frac{1}{V^2/2\frac{\partial\rho/\partial h}{\rho}\cos\sigma_{QEGC}}\frac{1}{C_L/C_D} \tag{5.39}$$

$$\dot{h}_{QEGC} = V\sin\gamma_{QEGC} \tag{5.40}$$

These equations can be used to modify the controls as

$$L\cos\sigma_c = L\cos\sigma_{nom} - k\left(\dot{h} - \dot{h}_{QEGC}\right) \tag{5.41}$$

where $\dot{h}$ is the current altitude rate, and k is a gain, which can be assumed constant, or scheduled according to the velocity. In this way, it is possible to strongly reduce the oscillations associated with a hypersonic gliding vehicle, as it is possible to see in Figs. (5.8), and (5.9).

Figures 5.8 and 5.9 show the altitude for four different missions with and without the correction term coming from the application of the quasi-equilibrium glide condition represented by Eq. (5.39). One can see that the correction brings strong reductions of the oscillations, having as a result a much smoother entry flight. This results in a reduced thermal and mechanical stress on the vehicle, and thus in a lighter structure and a larger payload mass.

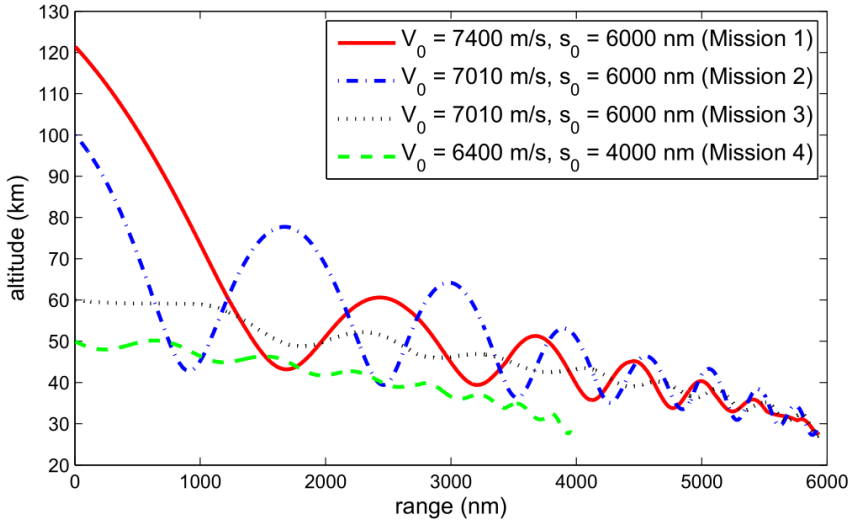


Figure 5.8: Altitude profiles for different missions without the quasi-equilibrium gliding condition correction[9].

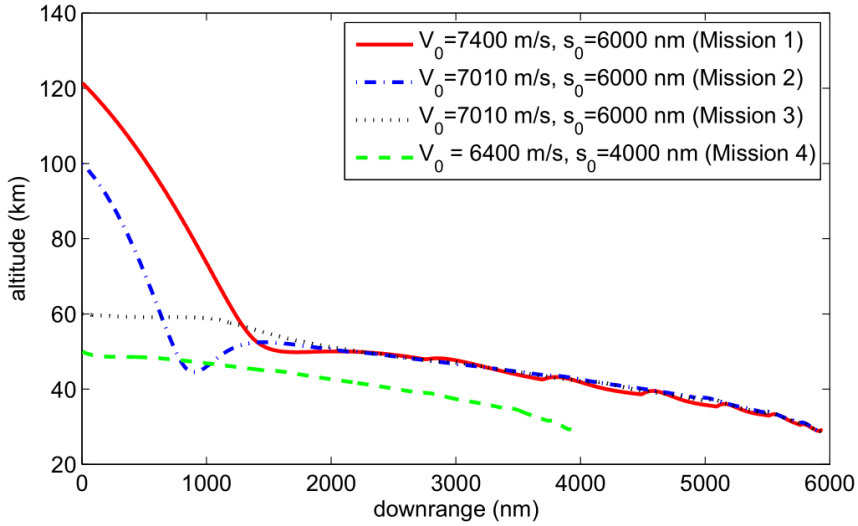


Figure 5.9: Altitude profiles for different missions with the quasi-equilibrium gliding condition correction[9].

5.6 Constraint-Tracking Approach

An alternative to drag-tracking methods has been proposed by Mooij[28]. With this approach a generalized constraint-tracking law is derived. As in the case of the drag-

based approaches, the scheme is composed of a constraint-based feedforward solution, and a tracking solution. For the latter, two alternatives are proposed: an output feedback tracking controller, and an adaptive output tracking, which will be described in the next sections.

5.6.1 Constraint-based Feedforward Guidance

The feedforward guidance solution based on constraint-tracking is of general application. In this model the following hypothesis are assumed to be valid.

- the Earth is non-rotating;
- there is a central gravity field;
- the atmospheric density ρ follows an exponential profile (i.e., the isothermal model is considered valid);
- α is given;

With these assumptions, one can demonstrate that

$$\frac{d(\rho V^n)}{dt} = \frac{d\rho}{dh}\dot{h}V^n + n\rho V^{n-1}\dot{V} = 0 \tag{5.42}$$

and the derivative of flight-path angle γ w.r.t. the velocity V can be written as

$$\frac{d\gamma}{dV} = \frac{1}{\cos\gamma}\left(\frac{nh_s c_{q_c}(2nKV^{n-1} + 2Kgh_s n(n-2)V^{n-3})}{(2KV^{n-2}(V^2 + gnh_s))^2}\right) \tag{5.43}$$

where:

- K is the ballistic coefficient, equal to $\frac{m}{C_D S}$, expressed in kg/m^2;
- c_{q_c} is the value of the constraint that the term ρV^n has to satisfy, and expressed in kg m$^{\text{n-3}}$/s$^{\text{n}}$
- h_s is the scale height, expressed in m;
- g is the gravity acceleration, given in m/s^2;

For $n = 2$ the guidance becomes a *g-load*-based guidance, while for $n = 6$, we have a *heat-flux*-based guidance. The bank angle can then be extracted from the differential equation of the flight-path angle γ.

$$\dot{\gamma} = \frac{d\gamma}{dV}\dot{V} \tag{5.44}$$

5.6.2 Output Feedback Tracking

In the output feedback tracking, a control law is built according to

$$\mathbf{\Delta u} = -\mathbf{K}\mathbf{\Delta y} \tag{5.45}$$

where $\mathbf{\Delta y}$ and $\mathbf{\Delta u}$ are the output we want to track, and the control effort, respectively. The output tracking variables proposed are the velocity V, and the pseudo-altitude h^*,

$$h^* = h + K_\gamma \gamma \tag{5.46}$$

where the gain K_γ is a gain scheduled according to the velocity modulus V. The error dynamics can be formulated as

$$\mathbf{\Delta \dot{x}} = \mathbf{A}\mathbf{\Delta x} + \mathbf{B}\mathbf{\Delta u} \tag{5.47}$$

and the terms $\mathbf{\Delta u}$ are the variations w.r.t. the nominal angle of attack, $\Delta\alpha$, and w.r.t. the nominal bank angle, $\Delta\sigma$. The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ can be derived analytically or numerically. The gain matrix $\mathbf{K}$ can be instead computed by solving a Riccati equations, once that the weight matrices $\mathbf{R}$ and $\mathbf{Q}$, corresponding to the control effort, and the maximum allowed output error, are set.

5.6.3 Adaptive Output Tracking

Alternatively, one can implement the so-called Model Reference Adaptive Guidance (MRAG)[45]. An example of the working scheme is shown in Fig. 5.10.

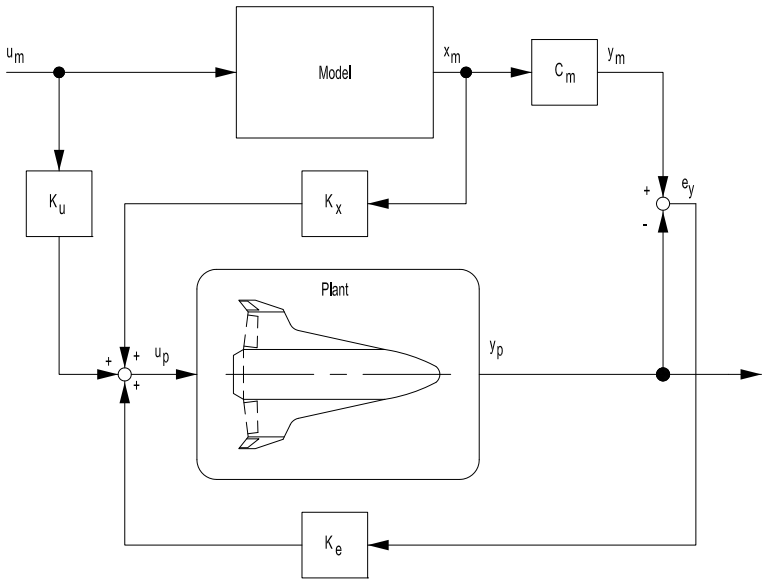


Figure 5.10: Model Reference Adaptive Guidance scheme[28].

The guidance law is given by

$$\mathbf{\Delta u}(t) = \mathbf{K_r}(t)\mathbf{r}(t) \tag{5.48}$$

where $\mathbf{r}(t) = [\mathbf{e_y}(t)\ \mathbf{x_m}(t)\ \mathbf{u_m}]$, and $\mathbf{K_r}(t) = [\mathbf{K_e}(t)\ \mathbf{K_x}(t)\ \mathbf{K_u}(t)]$. The model input $\mathbf{u_m}$ and model state $\mathbf{x_m}$, together with the so-called output error $\mathbf{e_y}$, will be part of the input signal $\mathbf{u_p}$ to the plant, while the gain matrices $\mathbf{K_e}(t)$, $\mathbf{K_x}(t)$, and $\mathbf{K_u}(t)$ are adaptive, and driven by the output error, the model state, and the model input, respectively. An example of performance of such a scheme is depicted in Fig. 5.11, where one can observe how the heat-flux is correctly tracked for almost 600 s, as expected. The two tracking

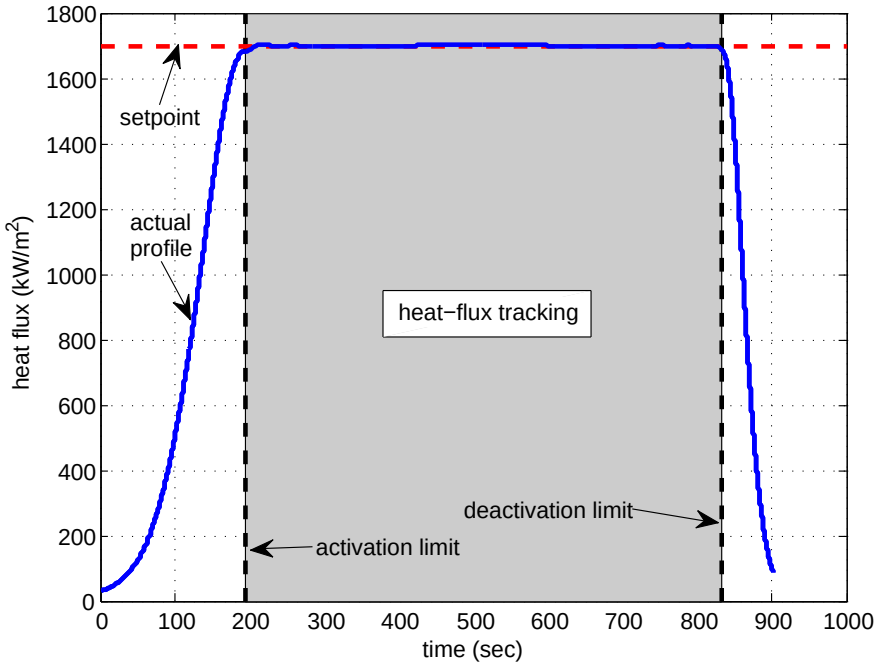


Figure 5.11: Heat-flux tracking guidance example[28].

approaches can be combined to take advantage of the benefits of both, providing a constraint-based integrated guidance scheme[28,45].

5.7 Optimal Control Approach

Ross and Fahroo introduced the pseudospectral optimal control theory[46,47], and showed how it can provide satisfying results with strongly nonlinear problems, such as the entry and the attitude reorientation of the International Space Station[5]. In this case, the reentry problem is formulated as an optimal control problem, which is efficiently transcribed in the form of standard nonlinear programming problem, and numerically solved. This is a process typically done offline.

To extend this approach, Bollino and Ross[22] demonstrated the potential use of these techniques on real-time systems, in virtue of the increased capabilities of the computational power over the last twenty years. Their idea is to solve the entry problem not only offline, but at each step with a given frequency. To show the results of their research, the software **DIDO**[6] was used for both the offline and online trajectory generation. The optimizer is then integrated in a scheme with a Runge-Kutta propagator, which provides the current state of the vehicle (Fig. 5.12).

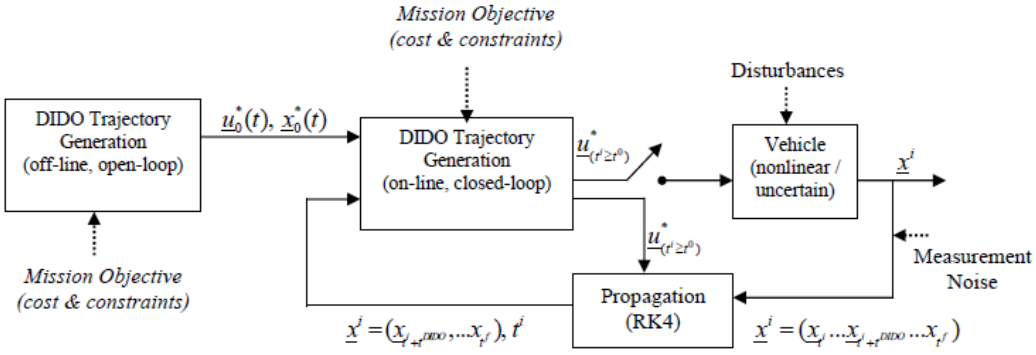


Figure 5.12: Optimal feedback control approach[22].

All constraints can be taken into account, together with the presence of eventual disturbances, during the generation of optimal trajectories. Figure 5.13 shows the results obtained for the X-33, in terms of position (subplot a), velocity (subplot b), controls (subplot c), and 3-D trajectory (subplot d). They satisfy all the requirements defined in terms of final position and velocity, as well as in terms of constraints, which have been included in the trajectory-generation process as algebraic constraints. The states are smooth, and the control limits are correctly taken into account.

After the generation of the nominal trajectory, wind gusts of different magnitude were included in the analysis. Moreover, effects such as the noise and the bias coming from the navigation system can be included in the analysis as well. From Fig. 5.14 we can conclude that, even in the presence of strong wind gusts, the spacecraft is able to reach the prescribed final position, while in case of open loop a significant deviation from the desired final altitude appears (the vehicle would have crashed down).

Specifically, in Fig. 5.14, the dotted line shows the altitude evolution over time in open loop without any disturbances. The final prescribed altitude, equal to 0 ft, is reached. This approach is not valid anymore when wind is included (dashed line): the vehicle will fly a significantly different trajectory with a potentially disastrous result (in the example the vehicle falls down more rapidly and the final desired position is completely missed). The introduction of the closed loop will make the reachability of the final position still possible even in presence of significant wind disturbances (continuous line). It is worth mentioning that this will be very likely a main direction to follow in the

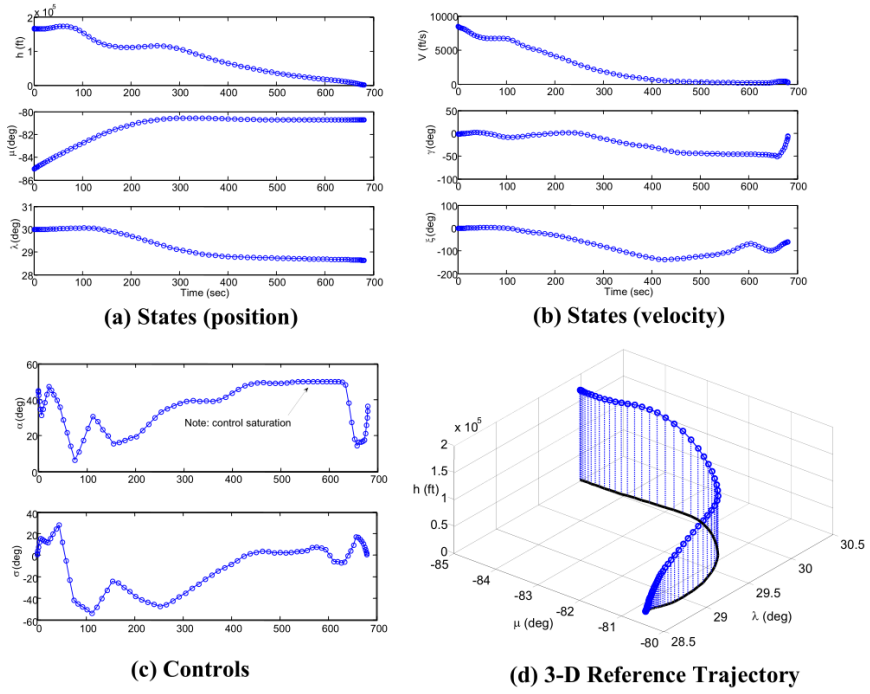


Figure 5.13: Optimal feedback control results obtained with DIDO for the X-33 [22].

future, as it provides reduced pre-analysis of the scenario, and larger flexibility during the mission. Anyway, for systems which cannot rely on the required CPU capabilities for the application of this method, alternatives must be conceived. This will be the subject of Chapter 8.

5.8 Further methods

So far we have briefly reported only some of the state-of-the-art techniques for the entry guidance problem. For instance, among the tracking methods, other techniques like Model Predictive Control [48] or H_∞ [49] can be used. However, in the general case, nonlinear MPC requires to solve a complete NLP iteratively at each step. H_∞ controllers work very well [50], but they are intrinsically linear, and require a gain-scheduling to be used on nonlinear systems. The same problem affects more traditional PID controller and LQR [16], which need to be properly sampled along the nominal trajectory. Over the years, several stimulating alternatives were explored, such as the use of on-board trajectory generation algorithms [51,52], or Sliding Mode Control [53,54]; these families of methods were considered suitable for SHEFEX-3, given their small computational burden, and at the same time, the good performance they provide in presence of uncertainties.

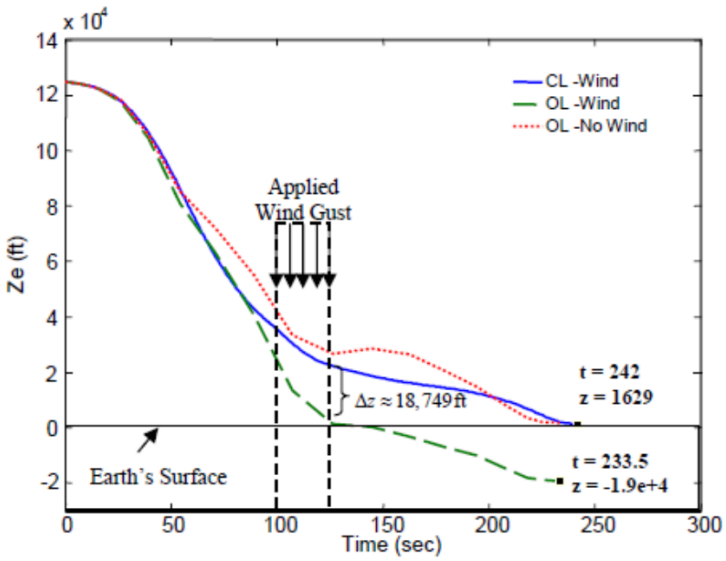


Figure 5.14: Performances of the open-loop and the closed-loop system in presence of strong wind gusts[22].

These methods, and how they have been adapted to SHEFEX-3 will be described in detail in Chapters 9 and 10. Moreover, it was implicitly supposed that the translational dynamics can be decoupled from the attitude dynamics, but they can also joined into a unified entry guidance approach. Bollino[55] extended his method to this scenario too, but this approach is out of scope of this work. Therefore, for SHEFEX-3, the inner and outer loop are completely separated, and for this reason, only the methods associated with the translational dynamics have been briefly reported here.

Chapter 6

Verification and Validation

The objective of this chapter is to describe the verification and validation logic followed during the development of the simulation tool. Activities of testing, validation and verification are a crucial aspect for the development of every algorithm, or more in general, of software. Specifically, the verification aims to check that the code was programmed in the right way. The validation aims to check that the programmed code fulfills its scope. In this specific case, not all the reference data is publically available, therefore, what is done in some cases is a pseudo-validation, but for convenience we will refer to this activity as validation. Finally, given the extension of the work, it is important to mention that only some of the performed tests are reported here. Specifically we will describe some of the tests and the V&V procedures for the following sub-modules:

- Atmosphere;
- Gravity;
- Aerodynamics;
- Equations of motion;

6.1 Atmosphere

As already stated, two models have been mainly used. The US76 model and the NRLMSISE-00 model.

6.1.1 US Standard Atmosphere 1976

The implemented model has been verified w.r.t. to the official US76 model look-up tables released in 1976[30]. Results of the comparison from 0 to 100 km with intervals of 10 km are listed in Table 6.1. The results are fully consistent. The maximum discrepancy between the models is 0.0175%, both for the temperature and the atmospheric density.

Table 6.1: Validation of US-76 model.

Altitude (km)	T	T_{ref}	ρ	ρ_{ref}
0	288.150	288.150	1.2250	1.2250
10	223.263	223.252	0.4135	0.4135
20	216.650	216.650	0.0889	0.0889
30	226.508	226.509	0.0184	0.0184
40	250.325	250.350	0.0040	0.0040
50	270.650	270.650	0.0010	0.0010
60	247.064	247.021	0.0003	0.0003
70	219.599	219.585	8.2822e-05	8.2829e-05
80	198.656	198.639	1.8456e-05	1.8458e-05
90	186.867	186.870	3.4102e-06	3.4160e-06
100	195.081	195.080	5.5942e-07	5.6040e-07

6.1.2 NRLMSISE-00

For this module the Mathworks validated block *NRLMSISE-00* belonging to the Aerospace blockset, and depicted in Fig. 6.1 has been used. Therefore, no further validation actions are required.

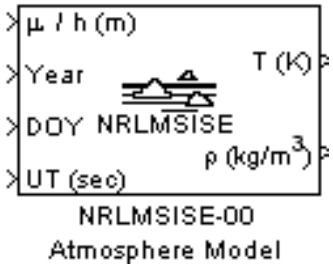


Figure 6.1: Mathworks NRLMSISE-00 block.

6.2 Gravity

As stated in Sec. 3.2, a simplified gravitational model having the J_2 contribution has been used. The coded model has been compared with the full WGS84 implemented in Matlab[56]. Figure 6.2 shows the gravity profiles (the continuous line is the gravity model used in this work, and the dashed line the full WGS84 model) obtained along a trajectory which goes from 100 to 0 km, longitude from -11.60 deg to 45.75 deg, and latitude from 71.89 deg to 66.40 deg, as described in Table 2.3. In the second plot, the difference between the two gravity profiles is depicted. For easiness of visualization, only the altitude is shown as independent variable.

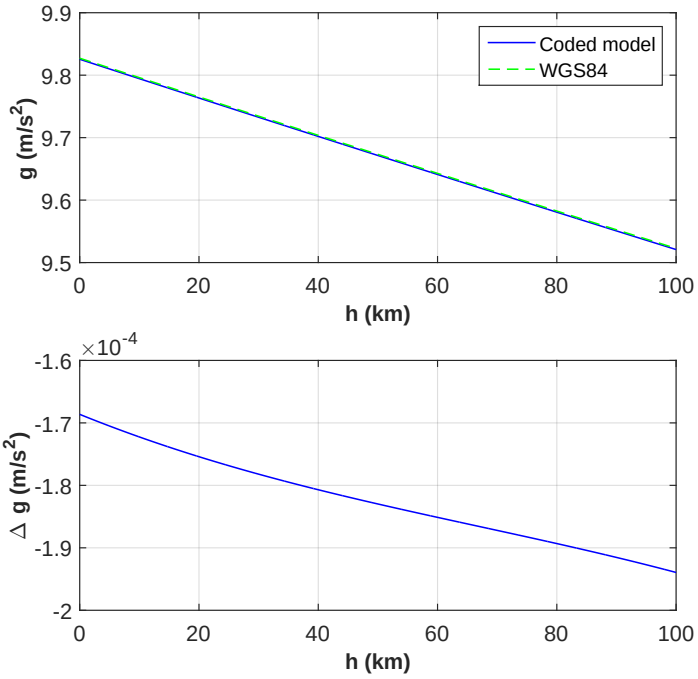


Figure 6.2: Discrepancy between simplified gravity model and WGS84.

The error is in the order of 0.2 mm/s^2, and for this type of mission, characterized by having a relatively short timeframe, can be considered fully acceptable.

6.3 Wind

For this module the Mathworks validated block *HWM07* belonging to the Aerospace blockset, and depicted in Fig. 6.3 has been used. Therefore, no further validation actions are required.

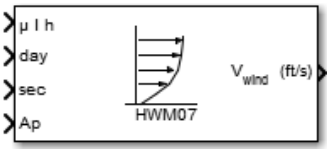


Figure 6.3: Mathworks HWM07 block.

6.4 Aerodynamics

The aerodynamics has been mutually validated with the DLR's system analysis and space transportation department, and the GNC experts of Airbus Defence and Space. Table 6.2 shows some of the values obtained during the analysis, which consists in implementing the procedure described in Sec. 3.4 to compute the C_L, C_D, and the flap angle τ. The first column and the first row indicate the angle of attack, and Mach number, respectively. The resulting coefficients have been enclosed in a Simulink block. The inputs of this block are α, M, and h, and the output are aforementioned C_L, C_D, τ, as well as all their derivatives w.r.t. the input variables.

Table 6.2: Validation of aerodynamics - inviscid C_L.

AoA (deg) / Mach	2	6	10
10.0	0.4256	0.3662	0.3522
12.5	0.6163	0.5136	0.4902
15.0	0.8158	0.6550	0.6224
17.5	1.0184	0.7912	0.7504
20.0	1.2164	0.9236	0.8770
22.5	1.4037	1.0526	1.0030

The three independent analyses gave exactly the same results, since the aerodynamic coefficients and the flap deflections were identical.

6.5 Equation of motion

To verify the correct behavior of the equations of motion, several tests have been conceived and performed. Some of the tests are reported here. The tests reported are

- Vertical motion without lift and drag;
- Elliptical motion without lift and drag;
- Periodic motion without drag;
- Symmetrical motion with lift and drag;
- Motion with and without Earth's rotation;

6.5.1 Vertical motion without lift and drag

In this first example the lift and the drag accelerations are shut down. The motion begins with flight-path angle equal to -90 deg. The velocity azimuth angle is not defined, as there is no horizontal component of velocity. The initial velocity modulus is equal to 1 m/s, while the gravity is kept constant, with a value of 9.7823 m/s^2. The vehicle is located at the altitude of 100 km, at zero longitude and zero latitude. Since the vehicle

is only subject to a constant gravity acceleration, it is possible to compute the time needed to fall to the sea-level altitude. If we integrate twice the equations of motion, we have

$$\begin{aligned} \ddot{h} &= -g \\ \dot{h} &= -gt - v_0 \\ h &= -g\frac{t^2}{2} - v_0 t + h_0 \end{aligned} \tag{6.1}$$

We can solve the third equation w.r.t. time to obtain the time of flight associated with an altitude variation from 100 to 0 km.

$$t_{TOF} = \frac{-v_0 \pm \sqrt{v_0^2 + 2gh_0}}{g} \tag{6.2}$$

and if we take the solution associated with the + sign, we find that the predicted time of flight is 142.8836 s.

Figures 6.4 shows the states obtained by propagating Eq. (4.7), without lift and drag accelerations.

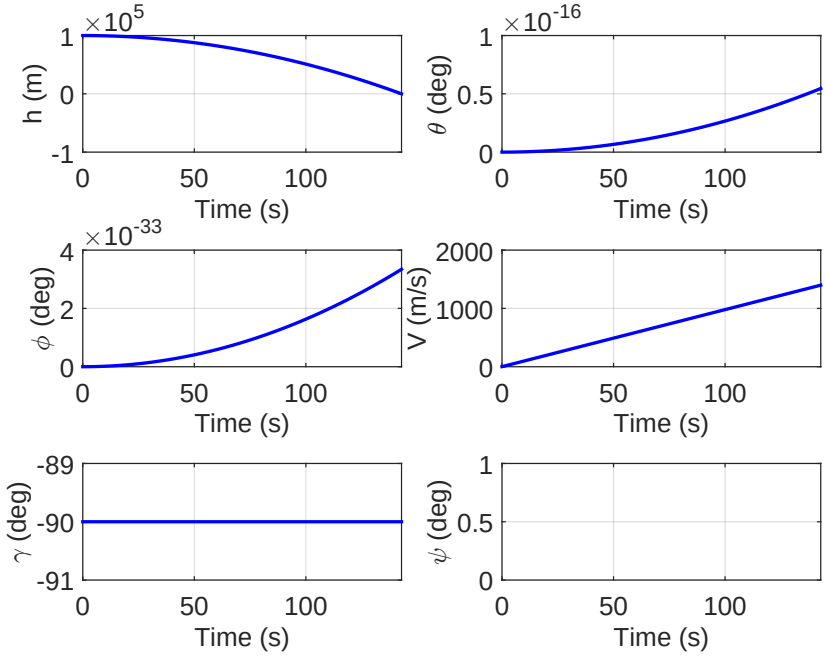


Figure 6.4: Vertical motion - states.

The motion follows a free fall as expected. The velocity increases linearly, in virtue of the constant gravity field. The flight-path angle remains constant, and so do the longitude and the latitude. The altitude decreases quadratically, as expected, according to Eq. (6.1). Finally, the velocity azimuth is not defined, as its differential equation

becomes singular in case of vertical motion, as one can easily check by looking at Eq. (4.7). Indeed, it is not even possible to properly talk about velocity azimuth angle when no horizontal component of velocity is defined, as the motion is orthogonal to the local North. The time of flight obtained with the simulation is 142.8836 s, which perfectly matches with the theoretical predictions.

6.5.2 Elliptical motion without lift and drag

Let us consider a different example. Also in this case lift, drag, and Earth's rotation are excluded. The vehicle flies again at an initial altitude of 100 km, starting with longitude and latitude equal to 0, the velocity-azimuth angle is equal to 90 deg, which means that the spacecraft flies towards East, and has an initial flight-path angle equal to 0. The initial velocity is equal to 7916.3 m/s (i.e., $1.01\sqrt{g_0 r_0}$), to induce an elliptical orbital motion, and no aerodynamics effects are included. The resulting ellipse has an eccentricity equal to about 0.02, with the apogee at about 365 km. The results of the propagation of the motion are depicted in Fig. 6.5.

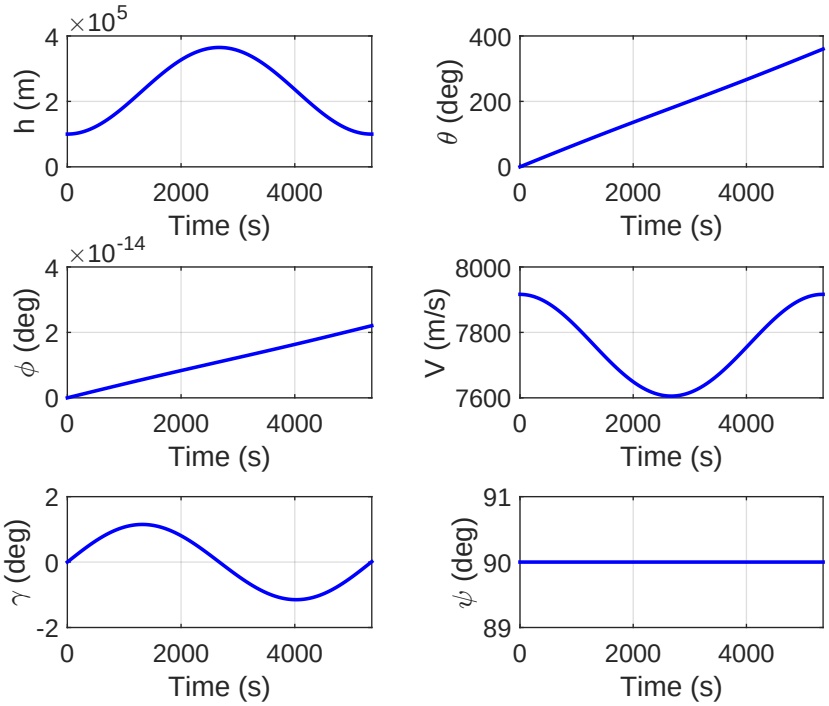


Figure 6.5: Elliptical motion - states.

The vehicle acts like a satellite, as expected. Since the drag is set to 0, the mechanical energy of the system remains constant.

$$\dot{E} = -DV = 0 \tag{6.3}$$

The flight-path angle oscillates and becomes equal to 0 at the perigee and at the apogee, while the velocity-azimuth angle remain constant. The longitude is a quasi-linear function of the time, while the latitude is constant. Finally, the velocity modulus and the altitude change according to the exchange of kinetic and potential energy, which in total remains constant, and is equal to -3.081·10^7 m^2/s^2.

6.5.3 Periodic motion without drag

Another interesting case is the free motion of the vehicle without drag, but with lift, and initial negative flight-path angle. The simulation is run for 1000 seconds. The initial flight-path angle is equal to -5 deg, the altitude is again 100 km, and the velocity is 6 km/s. The velocity-azimuth angle is equal to 0 deg, which means the vehicle flies towards the North. The initial latitude and longitude are again 0 deg. The angle of attack and the bank angle are constant, and equal to 30 deg and 0 deg, respectively.

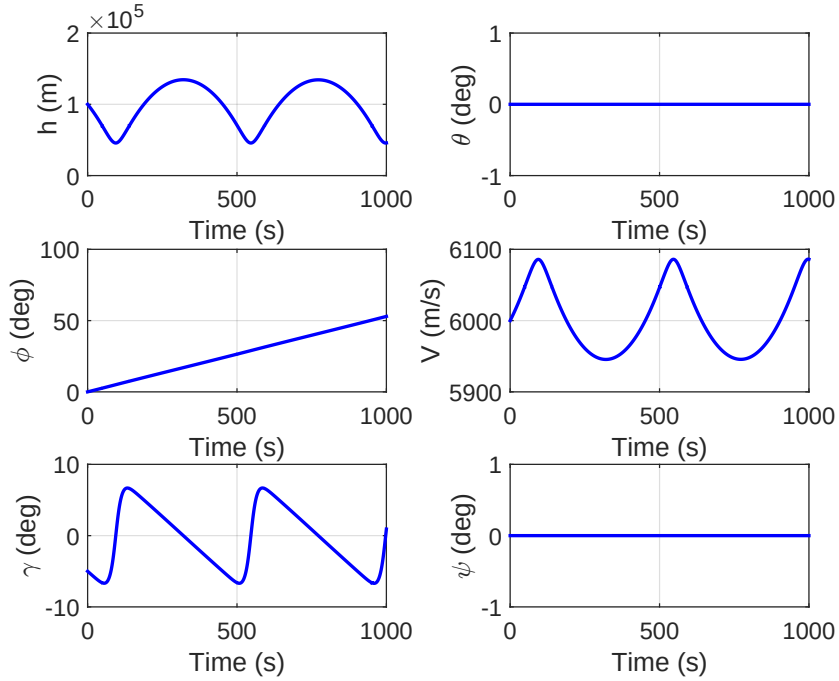


Figure 6.6: Periodic motion - states.

Since the mechanical energy is constant in this case too, there will be a continuous transformation of energy in its potential and kinematic forms, which will result into a periodic motion. This is exactly what the states in Fig. 6.6 and the trajectory in Fig. 6.7 show. The minimum values of altitude correspond to the maximum values of the velocity modulus, and viceversa. The longitude and the velocity azimuth angle remain constant, while the latitude increases, as expected.

Figure 6.7: Periodic motion - 3-D trajectory.

6.5.4 Symmetrical motion

In this example the symmetrical motion is considered. Here lift and drag are included in the analysis, while the Earth's rotation is neglected. The initial velocity is 4000 m/s, the flight-path angle is equal to 0, the velocity azimuth angle is 90 deg, which means that the vehicle flies towards East. Three cases are considered here. In the first the bank angle is kept constant to 30 deg; in the second it is kept to -30 deg, while in the last case it is simply equal to 0. Results are plotted in Figs. 6.8 and 6.9, where one can observe that the motion is symmetric when the bank angle is kept equal to 30 and-30 deg, respectively. Moreover, since the case with bank angle equal 0 deg has a larger component of lift directed upwards, the altitude for this case is larger than in the other two cases, as expected.

From the analysis of the states depicted in Fig. 6.8 we can see that when the bank angle is turned towards the right, the latitude decreases (continuous line), and the velocity azimuth angle, measured from North clockwise, increases. The opposite conclusions can be drawn for the dotted line. This behavior is fully in line with the expectations.

6.5.5 Motion with and without Earth's rotation

The last test reported here regards the influence of the Earth's rotation on the motion. In this case we consider an initial 120 km of altitude, with a velocity modulus of 7200 m/s. The initial flight-path angle is 0 deg, while the velocity-azimuth angle is 45 deg.

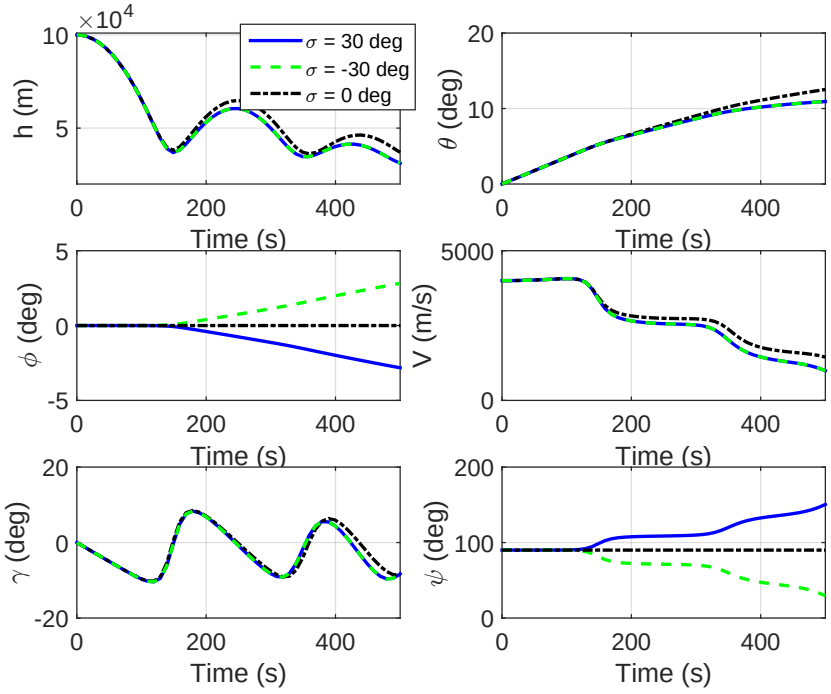


Figure 6.8: Symmetric motion - states.

The initial longitude and latitude are both 0 deg. The bank angle is kept equal to 0, and the angle of attack is kept constant, and equal to 30 deg. Two trajectories are computed, according to the models given by Eqs. (4.7) and (4.6). Neither lift nor drag are included in the analysis. Therefore, the only difference between the two trajectories is provided by the Earth's rotation $\omega_\oplus$. How can we quantify the difference? The difference in terms of final position can be expressed as

$$\Delta\mathbf{r} = \int_{t_0}^{t_F}\int_{t_0}^{t_F} \left[-2\omega_\oplus \times \dot{\mathbf{r}} - \omega_\oplus \times (\omega_\oplus \times \mathbf{r})\right] d\tau d\tau \tag{6.4}$$

where $-2\omega_\oplus \times \dot{\mathbf{r}}$ is the Coriolis acceleration (which is about 11% of gravity acceleration inter ms of modulus), while $\omega_\oplus \times (\omega_\oplus \times \mathbf{r})$ is the centrifugal acceleration, which is smaller (about 0.35% of g). The vectors $\mathbf{r}$ and $\dot{\mathbf{r}}$ are the vehicle's position and velocity vectors , while $\omega_\oplus$ is simply

$$\omega_\oplus = \begin{bmatrix} 0 \\ 0 \\ \omega_\oplus \end{bmatrix} \tag{6.5}$$

Since the vehicle flies in the Northern Hemisphere, the Coriolis acceleration will tend to deviate the trajectories to the right, while a component of it will be opposite to the gravity, leading to higher altitudes. States for the two cases are plotted in Fig. 6.10.

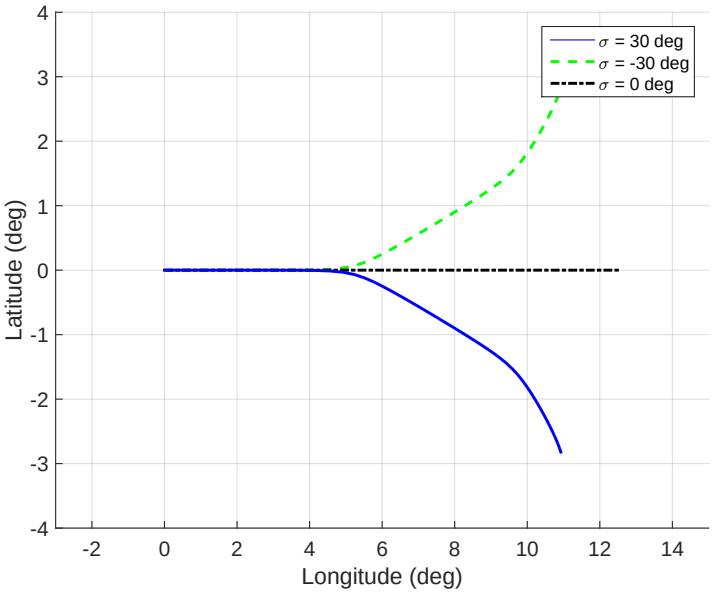


Figure 6.9: Symmetric motion - groundtrack.

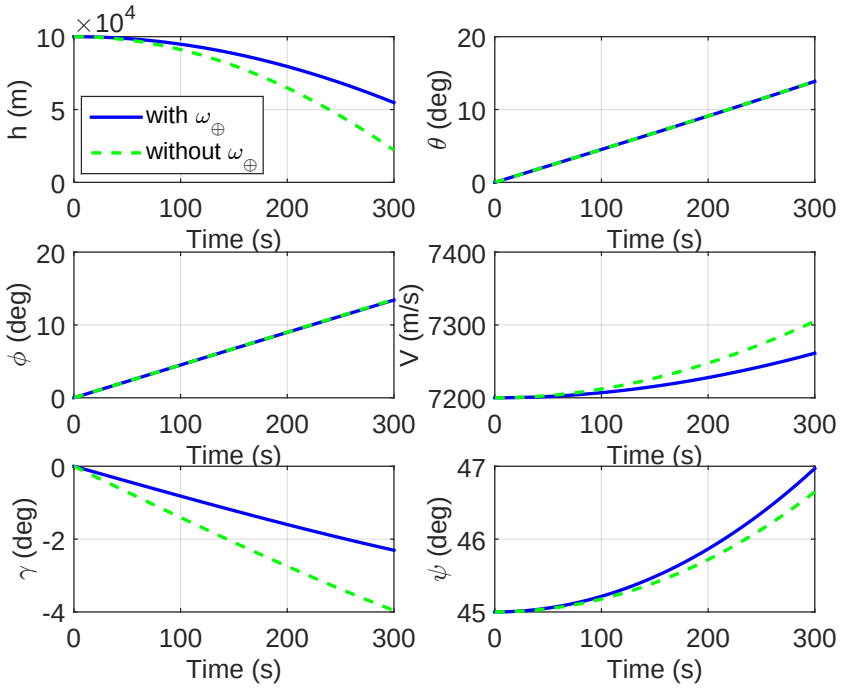


Figure 6.10: Motion with and without Earth rotation - states.

The double integration over time of the Coriolis acceleration gives a difference of 35.153 km, which is close to the difference between the two final positions (33.613 km), over a total range of about 2165 km. Looking at the states plotted in Fig. 6.10, (continuous lines) we can observe the deflection towards East in terms of velocity-azimuth angle, which, for the case with Earths rotation, becomes slightly larger, and therefore, closer to 90 deg, which means closer to the local East.

All the tests gave results consistent with expectations. Further validation (including entire trajectory comparisons) with DLR's system analysis and space transportation department, and with Airbus Defence and Space experts were done, but they have been omitted for brevity.

Part II

Guidance Development

Chapter 7

Generation of Optimal Trajectories

After having introduced the mission, the historical development of entry guidance, and the models for the environment, the dynamics, and the controls, it is now possible to focus on the guidance algorithms. This is the purpose of part II of the work. Specifically, in Chapters 7-9 the main guidance scheme adopted for SHEFEX-3 is explained in detail, while in Chapter 10 the method implemented as backup guidance is described. The Monte Carlo analysis of the developed methods is the subject of Chapter 11, while in Chapter 12 the main conclusions and lessons learned from this work are reported.

For what regards the main guidance, we can identify three main modules:

- Development of the optimizer;
- Development of the adaptive feedforward guidance;
- Development of the robust feedback controller;

The first part deals with the theoretical conception and the practical implementation of SPARTAN (SHEFEX-3 Pseudospectral Algorithm for Reentry Trajectory ANalysis), a tool developed for this specific work, but that was used for other scenarios as well [12,13]. Several practical implementation aspects, like scaling and Jacobian computation, are covered here. Moreover, the tool has been validated by comparison with several well-known examples from literature. With this approach, we are able to formulate the entry guidance problem as an optimal-control problem, which can be efficiently solved, and provides a trajectory corresponding to a specific initial-conditions set. In the second part the tool is coupled with an algorithm based on the so-called Adaptive Multivariate Pseudospectral Interpolation (AMPI). This gives us the possibility to generate an adaptive on-board trajectory, which initial states are different from the nominal conditions, but enclosed within some predefined limits. Given a trajectory-database, it is possible to select a subspace enclosing the real inflight-conditions. The selected subspace is represented by a hexeract, i.e., a hypercube having dimensions

equal to 6. By merging this logic with pseudospectral methods, it is possible to compute a complete feedforward guidance solution in an efficient way, as will be explained in Chapter 8. Once the feedforward guidance is properly defined, large initial dispersions can be handled. However, uncertainties in atmospheric density, mass, aerodynamics, as well as wind gusts, require the use of a feedback controller. Therefore, a dedicated novel sliding mode controller (SMC), described in detail in Chapter 9, has been implemented. This controller is able to deal with nonlinear systems, therefore no gain scheduling techniques need to be applied. It is computationally lighter and more efficient, since it is completely defined by a small number of constant parameters. The overall result is a system able to deal with multiple off-nominal conditions, uncertainties and disturbances, and which is able to generate stable and flyable trajectories until the TAEM interface is reached. The overall working scheme is shown in Fig. 7.1.

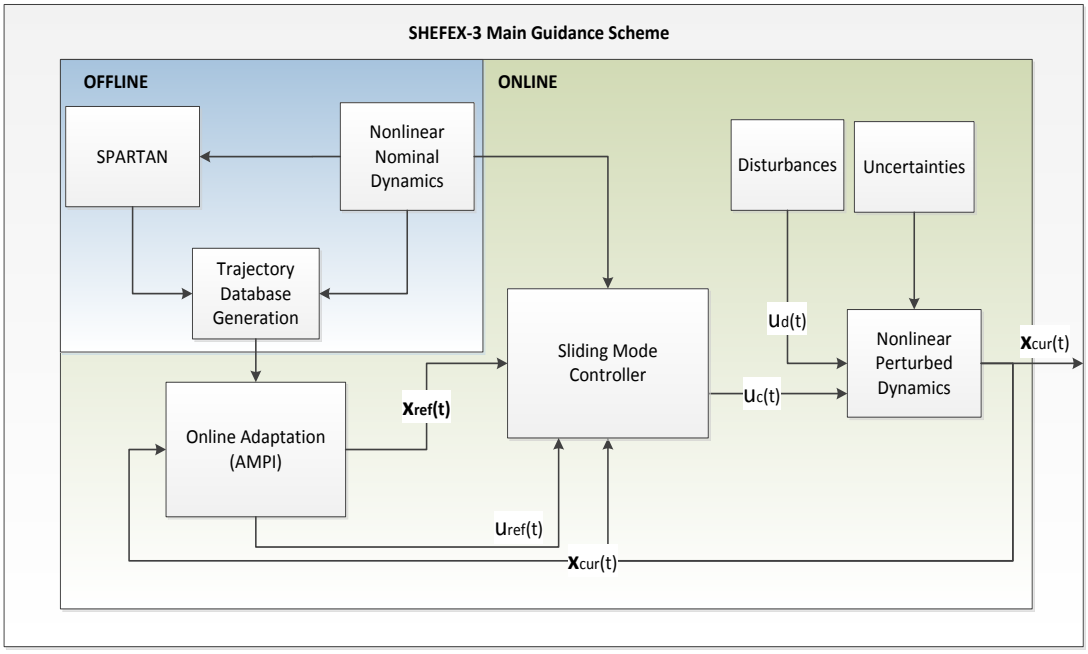


Figure 7.1: Main guidance scheme.

From the analysis of Fig. 7.1 we can distinguish an offline sub-module, which is run before the mission takes effectively place, and an online part. The generation of the trajectory database is clearly done offline by using SPARTAN. The full non-linear dynamics is taken into account, and the nominal atmospheric and aerodynamics models are used. The resulting database is stored in the on-board memory. According to the off-nominal conditions, the trajectory is online adapted through the previously cited AMPI technique. We will see that the use of this technique has the advantage to significantly reduce the memory requirements (up to 96% for the current scenario). The online adaptation is clearly performed in real time, and the feedforward controls are

combined with sliding mode control techniques to control the full nonlinear system in presence of multiple uncertainties and disturbances.

7.1 Definition of Optimal-Control Problem

There are several approaches for the generation of reference trajectories. Some methods exploit the structure of the specific problems we deal with, and are then particular solutions valid under given conditions. Another approach, which is gaining popularity, and is helped by the development of the computational capabilities of modern CPUs is the representation of the trajectory generation problem as optimal-control problem. This means we are looking for solutions minimizing (or maximizing) a given criterion and satisfying at the same time several constraints, which can be differential (e.g., the equations of motion of a spacecraft) and / or algebraic (e.g., the maximum heat-flux that a vehicle can tolerate). The standard form for representing optimal-control problems is the Bolza problem.

7.1.0.1 Bolza problem

Minimize (maximize) the cost function J

$$J = \Psi\left[\mathbf{x}\left(t_0, t_f\right)\right] + \int_{t_0}^{t_f} \Phi\left[\mathbf{x}(\tau), \mathbf{u}(\tau)\right] d\tau \tag{7.1}$$

subject to the differential equations

$$\dot{\mathbf{x}} = \mathbf{f}\left(t, \mathbf{x}, \mathbf{u}\right) \tag{7.2}$$

and to the path constraints

$$\mathbf{g}_L \leq \mathbf{g}\left(\mathbf{x}, \mathbf{u}\right) \leq \mathbf{g}_U \tag{7.3}$$

The Mayer term in the cost function (7.1) represents punctual constraints (e.g., the minimization of a distance according to a given metric), while the Lagrange term is usually referred to maximize or minimize variables over the entire mission (e.g., the heat load obtained by integrating the heat-flux over time).

In general, we deal with problems having bounded states and controls.

$$\mathbf{x}_L \leq \mathbf{x}(t) \leq \mathbf{x}_U \tag{7.4}$$

$$\mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U \tag{7.5}$$

What has been defined in Eqs. (7.1)-(7.5) represents a generic continuous optimal control problem. In the next section we will see what are the possible approaches to solve it.

7.2 Direct Methods

The continuous problem defined in Section 7.1.0.1 can be theoretically solved using calculus of variations and the application of Pontryagin's minimum principle[57]. This approach is hardly feasible in practice, because of the complexity of the equations. Instead, the problem can be treated very well numerically. This means that the problem needs to be discretized, or more specifically, *transcribed* in a proper way. The transcription process strongly depends on the chosen method. In all of the cases, the controls are computed over a discretized domain, while the states can be discretized or not, according to the specific methods implemented. For instance, multiple shooting can have discretized controls only, or discretized states and control[58]. In this work, the reference algorithm adopted for the transcription process is the flipped Radau pseudospectral method[46,47,59,60].

7.2.1 Pseudospectral Methods

Pseudospectral methods belong to the class of direct methods, meant as a family of algorithms, which objective is a clear and efficient discretization of the original optimal-control problem. The objective of the transcription is indeed to define a new, equivalent problem, having a finite number of variables and constraints, called Nonlinear Programming (NLP) Problem, which solution approximates the continuous one. Among the different direct methods, pseudospectral methods are gaining popularity after their successful application to the reorientation of the ISS with the zero-propellent maneuver[5]. The idea behind these methods is the use of a particular set of nodes where to discretize the states and the controls. It is possible to observe that the choice of these points has the remarkable advantage to completely remove the Runge phenomenon[1], without the use of more complex interpolation techniques such as B-splines[59].

7.2.2 SPARTAN

For the realization of this work, a dedicated software, SPARTAN (SHEFEX-3 Pseudospectral Algorithm for Reentry Trajectory ANalyis) implementing this method, has been developed[14,15]. All the optimal trajectories presented in this work have been generated using SPARTAN. The software includes advanced features, such as the exploitation of the Jacobian matrix of the discrete problem, and nonlinear automatic scaling techniques. These sub-modules and their utility will become clearer in the next sections.

7.2.3 Transcription Process

We can identify 5 different phases, which are part of a transcription method. They are:

1. The determination of the discrete domain.

[1]Runge phenomenon is a problem of oscillations at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree over a set of equispaced interpolation points.

2. The characterization of the differential and the integral operators.

3. The scaling procedure of the NLP problem.

4. The computation of the Jacobian matrix of the NLP problem.

5. The conversion of the solution of the NLP problem into a continuous form.

The phases of the transcription process are depicted in Fig. 7.2. Once we have chosen the number of nodes we want to use to transcribe the OCP, the discretization of the domain begins. The number of nodes needed to properly represent the OCP depends on the nature of the problem we are dealing with.

For the cases analysed here, 75 nodes were considered to achieve a good compromise between the accuracy of the solutions, and the size of the trajectory database. However, SPARTAN was tested with several problems, and with a varying number of nodes. For lunar landing guidance-problems 30 nodes were in many cases sufficient[13]. In testing phases, simulations up to 600 nodes have been performed without any problem[11]. The number of nodes directly affects the size and the computation of the differential and integral discrete operators, as we will see. Phases 1 and 2 are therefore mandatory. Without these modules, it is not possible to talk about a transcription. Phases 3 and 4 are usually included in the NLP solvers, and are based on the use of isoscaling / Jacobian rows normalization[58] and finite differences schemes[61], respectively. However, better results can be obtained by direct implementation of more sophisticated routines for the scaling and the Jacobian computation of the NLP problem.

The final result is the complete description of the discretized form of the continuous OCP, that is, the NLP to be solved, which can be provided to one of the well-known NLP solvers available off the shelf[61,62]. Indeed, the NLP solver is considered an external module with respect to the transcription, and for this reason, the solving phase has not been explicitly included in the phases cited in Sec. 7.2.3, and in the diagram of Fig. 7.2. Specifically, in all of the cases included in this work, SNOPT[61] has been used. Once the NLP problem has been solved, the results can be converted into a continuous form (Phase 5), to conclude the transcription process. In the following paragraphs, the implementation of each of these five phases will be explained more.

7.2.4 Determination of the discrete domain

The first step is the selection of a finite number of points which represent the discrete domain of definition for the Nonlinear Programming Problem. The continuous states $\mathbf{x}(\mathbf{t})$ and controls $\mathbf{u}(\mathbf{t})$ involved in the optimal-control problem are replaced by sets of discrete variables.

$$\mathbf{x}(t) \rightarrow [\mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_N] \tag{7.6}$$

$$\mathbf{u}(t) \rightarrow [\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_N] \tag{7.7}$$

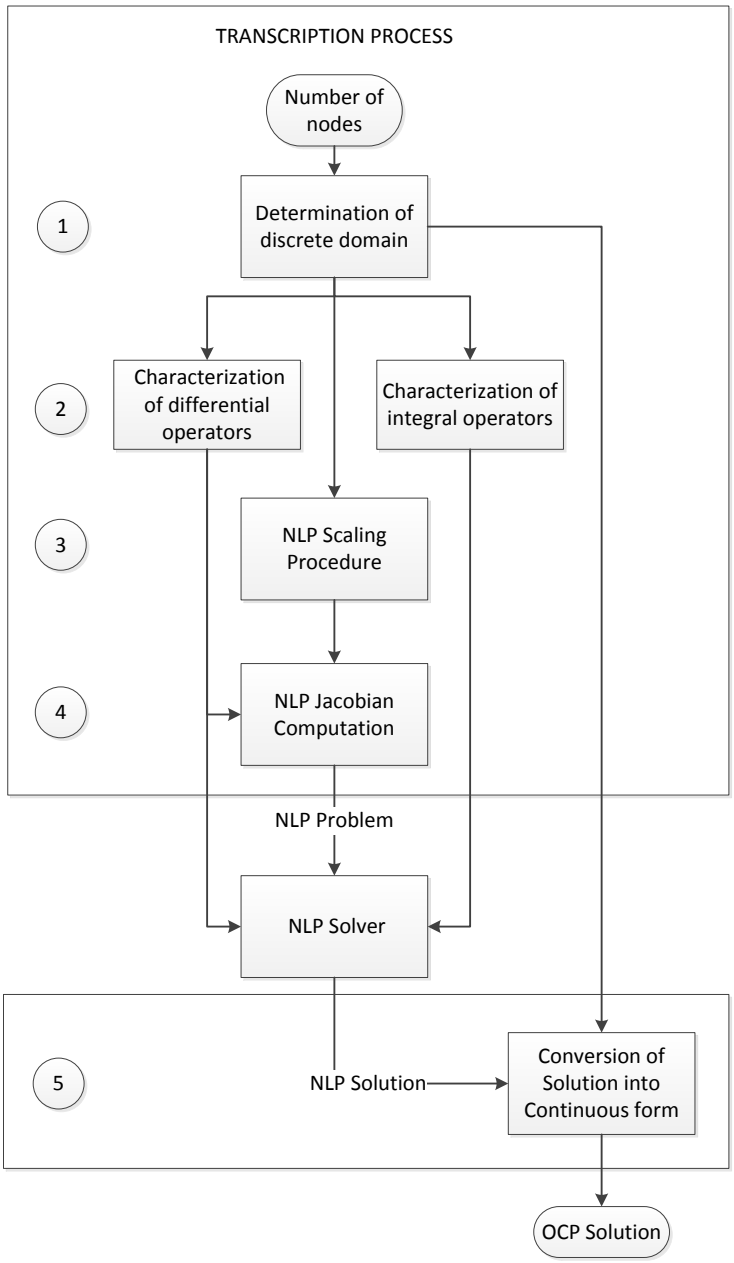


Figure 7.2: SPARTAN transcription process.

We can see a first characteristic of the flipped Radau pseudospectral method. The states are discretized using $N+1$ nodes, while the controls using N nodes. For this reason, we will refer to the $N+1$ nodes as *discretization nodes*, while the N nodes associated with the controls are usually called *collocation nodes*. The name refers to

the fact that, as we will see in the next sections, the differential equations and the algebraic constraints are *collocated* on the N nodes, even if they will be computed using the information extracted by the discretized $N+1$ nodes. Obviously, the set of discretization nodes is always the union of the collocation nodes and the initial point associated with the initial state $\mathbf{X}_0$. The initial control $\mathbf{U}_0$ is usually considered known, or can be extrapolated by using Eq. (7.12) evaluated at $t = t_0$. Accordingly to the states and the controls, the time needs to be discretized.

$$t \to [t_0, t_1, ..., t_N] \tag{7.8}$$

For this class of methods, the isolation property is valid, that is,

$$\mathbf{x}(t_k) = \mathbf{X}_k = \mathbf{X}(t_k),\ k \in [0, N] \tag{7.9}$$

$$\mathbf{u}(t_k) = \mathbf{U}_k = \mathbf{U}(t_k),\ k \in [1, N] \tag{7.10}$$

These discrete points can be used to build polynomial functions approximating the original continuous variables. This can be done using the Lagrange polynomial representation. This approximation allows to evaluate the discrete solution in any point belonging to the continuous time domain $t \in [t_0, t_f]$. The Lagrange polynomial approximating the continuous function will then be

$$\mathbf{x}(t) \cong \sum_{i=0}^{N} \mathbf{X_i} P_i^x(t) \tag{7.11}$$

$$\mathbf{u}(t) \cong \sum_{i=1}^{N} \mathbf{U_i} P_i^u(t) \tag{7.12}$$

where $P_i^x(t)$ and $P_i^u(t)$ are defined as

$$P_i^x(t) = \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{t - t_k}{t_i - t_k} \tag{7.13}$$

$$P_i^u(t) = \prod_{\substack{k=1 \\ k \neq i}}^{N} \frac{t - t_k}{t_i - t_k} \tag{7.14}$$

From Eqs. (7.11), and (7.12) it is possible to see that the Lagrange polynomials combined with the discrete states $\mathbf{X}_i$ and controls $\mathbf{U}_i$ will provide the approximated representations of the original continuous functions. Indeed, they will be the basis for obtaining the differential operator needed for the correct discretization of the differential equations defined in Eq. (7.2). The question is then how to choose the terms t_k that

appear in Eqs. (7.13), and (7.14). Historically, the first step for approximating a continuous domain has always been a uniformly distributed series of nodes. Pseudospectral methods use roots of Legendre polynomials instead, or more generally, roots of linear combinations of Legendre polynomials of different orders. Specifically, let us define the generic Legendre polynomial of order N[63].

$$L_N(\tau) = \frac{1}{2^N N!} \frac{d^N}{d\tau^N} \left[(\tau^2 - 1)^N \right] \tag{7.15}$$

The variable τ is defined over the domain $[-1, 1]$. In the flipped Radau pseudospectral (FRP) method, the collocation points are selected as the roots of the polynomial R_N, computed as a linear combination of the Legendre polynomials of order N and $N-1$ with coefficients 1 and -1.

$$R_N(\tau) = L_N(\tau) - L_{N-1}(\tau) \tag{7.16}$$

If the coefficients had been -1 and 1 instead, the method would have become the direct Radau pseudospectral method, which relies on the assumption that the final state is known, and the initial state unknown[59]. For our purpose, the flipped Radau pseudospectral method is the most natural choice, as we are supposed to always work with problems having known initial states.

An example of Radau collocation points is shown in Fig. 7.3, where $N = 11$ nodes is considered, together with the uniform distribution having the same number of nodes. One can see that the points are exactly roots of the Legendre-Radau polynomial. An important property of these polynomials is that over the interval they are orthogonal. This means that

$$\int_1^1 P_m(x) P_n(x) dx = \frac{2}{2n+1} \delta_{m,n} \tag{7.17}$$

where $\delta_{m,n}$ is the Kronecker delta.

At first glance, the Lagrange polynomials can be built using every criterion for the choice of the nodes, for instance a uniform distribution. So why choosing the roots of the Legendre polynomials instead? The answer resides in the following property: the distribution of points derived from the roots of this particular family of functions is not affected by the *Runge phenomenon*, an oscillatory behavior which does not permit a good approximation of the interpolating function in the extremes of the interval considered. To better understand this phenomenon, let us consider the function plotted in Fig. 7.4.

$$F(\tau) = \frac{1}{1 + 25\tau^2} \tag{7.18}$$

and let us sample it at 11 uniformly distributed nodes. In this example $F(\tau)$, defined in the interval $[-1, 1]$ represents the function we want to rebuild via interpolation using a discrete set of values $F(\tau_i)$, $i = 0, 1.., 10$. With the chosen distribution of points, the Lagrange polynomial will show a poor approximation, as anticipated, in the extremes of the interval considered, as we can see in Fig. 7.5 (a).

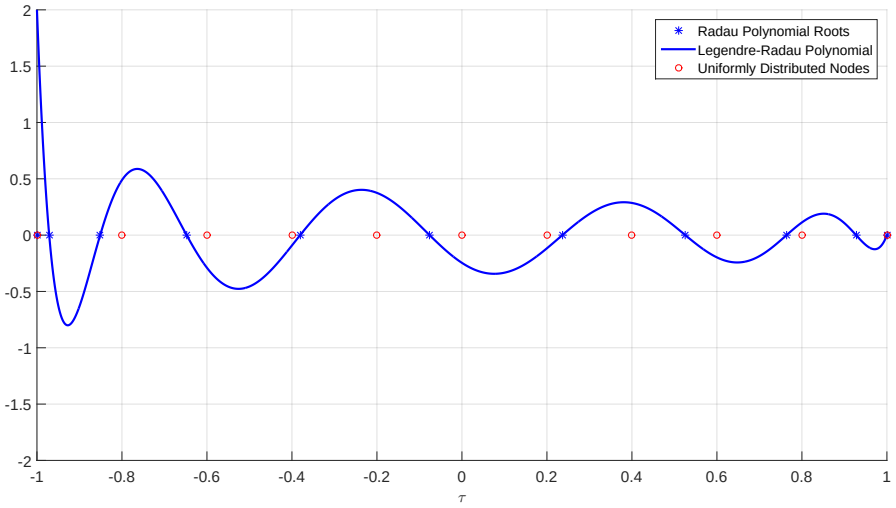


Figure 7.3: Collocation nodes using Radau polynomials roots and uniformly distributed nodes.

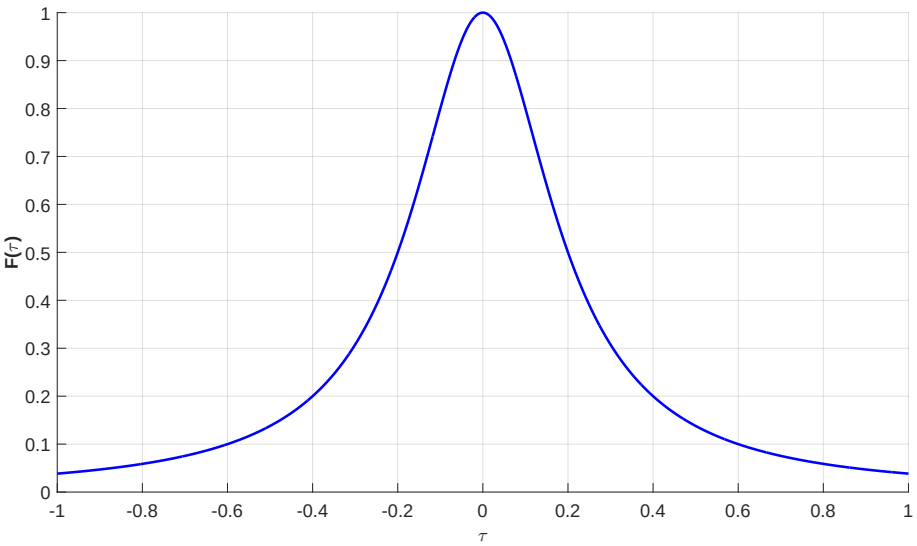
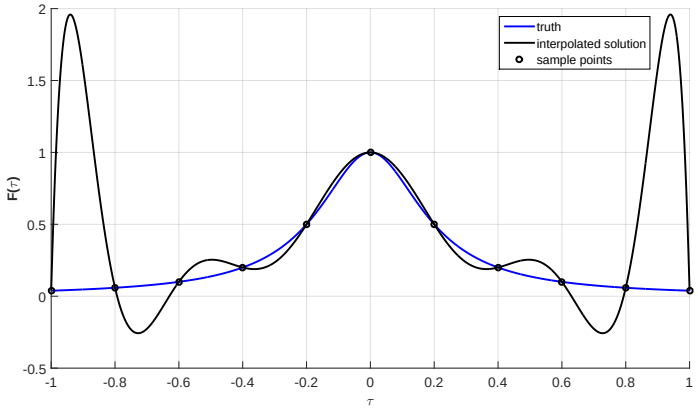
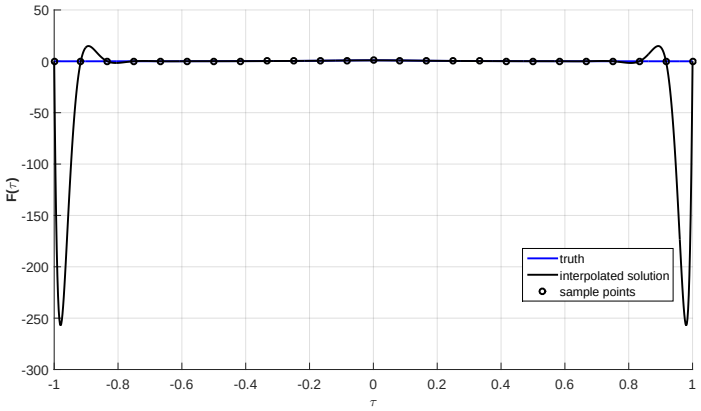


Figure 7.4: Runge phenomenon: test function $1/\left(1+25\tau^2\right)$.

Intuition would suggest to simply increase the number of nodes; as a consequence, the amplitude of the Runge oscillations become even larger, and the approximation worse, as shown in Fig. 7.5(b). Runge phenomenon can be avoided by other families of interpolation schemes have similar properties (i.e., different type of splines), but their



(a) Uniform distribution using 11 nodes.

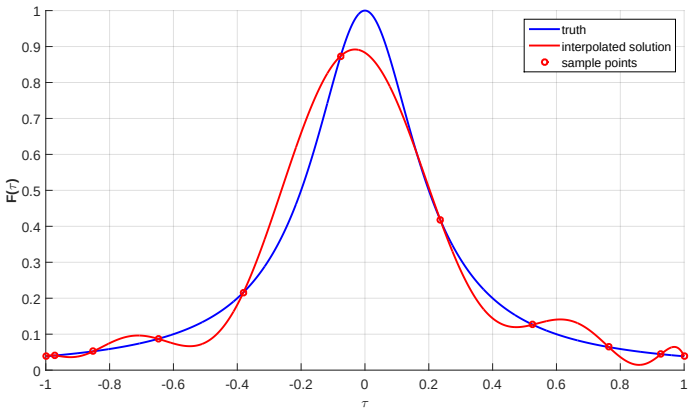


(b) Uniform distribution using 25 nodes.

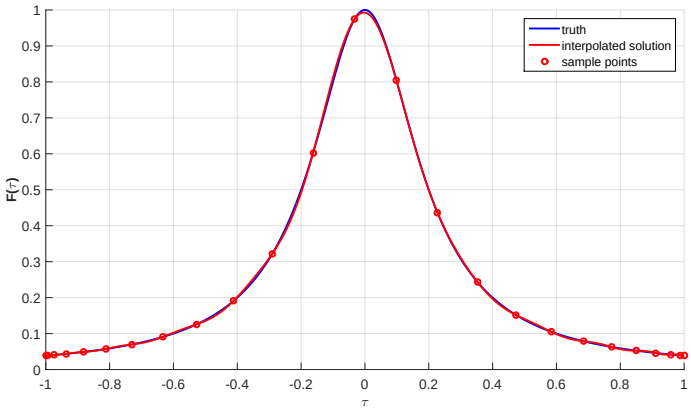
Figure 7.5: Runge phenomenon: interpolation using uniformly distributed nodes.

computation is heavier, and in some cases require the solution of a Linear Programming Problem. Figures 7.6(a) and 7.6(b) show how this phenomenon is completely avoided using the roots of the Legendre polynomials instead. The reason for this behavior resides in the fact that Legendre polynomials are orthogonal in the domain $\tau \in [-1, 1]$. We will refer to this domain as *pseudospectral time*. Since our problems are defined over a domain represented by the physical time, a mapping between the pseudospectral time $\tau \in [-1, 1]$ and physical time $t \in [t_0, t_f]$ is needed. It is immediate to demonstrate that this mapping can be built as two affine functions defined by Eqs. (7.19) and (7.20):

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2} \tag{7.19}$$



(a) FRP distribution using 11 nodes.



(b) FRP distribution using 25 nodes.

Figure 7.6: Runge Phenomenon: interpolation using distribution of nodes derived from Legendre polynomials.

$$\tau = \frac{2}{t_f - t_0} t - \frac{t_f + t_0}{t_f - t_0} \tag{7.20}$$

These properties, together with the ease of implementation, justify our choice for the characterization of the discrete domain.

7.2.5 Characterization of the differential and the integral operators

7.2.5.1 Differential Operator

Once the domain has been discretized, the differential operator needs to be derived. In other words, it is fundamental to find a good discretization of Eq. (7.2). The differential

operator will be replaced by a matrix relationship having the form given by:

$$\dot{\mathbf{X}}_i \cong \mathbf{D} \cdot \mathbf{X}_i, \ , i = 1, ...N \tag{7.21}$$

Then, the dynamics defined in Eq. (7.2) will be replaced by

$$\mathbf{D} \cdot \mathbf{X} = \frac{t_f - t_0}{2} \mathbf{f}(t, \mathbf{X}, \mathbf{U}) \tag{7.22}$$

In case of the flipped Radau method matrix the $\mathbf{D}$ has dimensions $[N, N+1]$. Again, this is due to the fact that the states are defined for $N+1$ discrete points, while the controls $\mathbf{U}$ and the derivatives of the states $\mathbf{f}(t, \mathbf{X}, \mathbf{U})$ are defined in the N collocation points. This means that the initial state $\mathbf{X}_0$ is an input and not an output of the optimization, and it is thus assumed to be known.

The term $\frac{t_f - t_0}{2}$ is a scale factor coming from the mapping between pseudospectral and physical time described by Eqs. (7.19) and (7.20). Our purpose is to derive the elements of the matrix $\mathbf{D}$. We can start from the definition of the Lagrange polynomial given in Eq. (7.11). Taking the derivative w.r.t. time, we get

$$\dot{\mathbf{x}}(t) \cong \frac{d}{dt} \sum_{i=0}^{N} \mathbf{X_i} P_i^x(t) \tag{7.23}$$

which yields

$$\dot{\mathbf{x}}(t) \cong \sum_{i=0}^{N} \mathbf{X_i} \frac{d}{dt} P_i^x(t) \tag{7.24}$$

as the nodal points are time-independent. Combination of Eq. (7.24) with (7.13) yields

$$\dot{\mathbf{x}}(t) \cong \sum_{i=0}^{N} \mathbf{X_i} \frac{d}{dt} \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{t - t_i}{t_k - t_i} \tag{7.25}$$

If we apply the rule of the derivative of product of functions N times to the i^{th} term of Eq. (7.25), we get

$$\frac{d}{dt} \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{t - t_i}{t_k - t_i} = \sum_{j=0}^{N} \frac{1}{t_j - t_i} \prod_{\substack{k=0 \\ k \neq i,j}}^{N} \frac{t - t_i}{t_k - t_i} \tag{7.26}$$

Then, combining Eq. (7.26) with Eq. (7.25), we have

$$\dot{\mathbf{x}}(t) \cong \sum_{i=0}^{N} \mathbf{X_i} \sum_{\substack{m=0 \\ m \neq i}}^{N} \frac{1}{t_m - t_i} \prod_{\substack{k=0 \\ k \neq i,m}}^{N} \frac{t - t_i}{t_k - t_i} \tag{7.27}$$

To transcribe the dynamics, we are interested to estimate the state derivative $\dot{\mathbf{x}}(t)$ only in the collocation points. Then the expression will become

$$\dot{\mathbf{X}}_\mathbf{j} \cong \sum_{i=0}^{N} \mathbf{X}_\mathbf{i} \sum_{m=0}^{N} \frac{1}{t_m - t_i} \prod_{\substack{k=0 \\ k \neq i,m}}^{N} \frac{t_j - t_i}{t_k - t_i},\ j = [1, ...N] \tag{7.28}$$

Finally, the elements of the matrix $\mathbf{D}$ in the Eq. (7.22) can be computed as

$$D_{j,i} = \frac{\partial \dot{\mathbf{X}}_j}{\partial \mathbf{X}_i} = \sum_{m=0}^{N} \frac{1}{t_m - t_i} \prod_{\substack{k=0 \\ k \neq i,m}}^{N} \frac{t_j - t_i}{t_k - t_i} \tag{7.29}$$

Equation (7.29) can be efficiently computed with the Barycentric Lagrange Interpolation[64]. In Fig. (7.7) it is possible to see how $\mathbf{D}$ can be used to estimate the derivative of a continuous function sampled in the collocation nodes. Eq. (7.30) defines a test function $F(\tau)$.

$$F(\tau) = Ae^{-\tau} \sin(\omega\tau) \tag{7.30}$$

Its derivative is numerically built using the differentiation matrix $\mathbf{D}$, and it can be seen that it fits the analytical solution very well. Moreover, they are integration schemes, which preserve the quadratic invariants of the system.

7.2.5.2 Integral Operator

The cost function in Eq. (7.1) may contain an integral term, which needs a proper discretization. In that case the Gauss quadrature formula is used[63]. This approach consists of replacing the continuous integral with the discrete sum given by:

$$\int_{t_0}^{t_f} \Phi\left[\mathbf{x}(\tau), \mathbf{u}(\tau)\right] d\tau = \frac{t_f - t_0}{2} \sum_{i=0}^{N} w_i \Phi\left[\mathbf{X}_i, \mathbf{U}_i(\tau)\right] \tag{7.31}$$

It can be shown that Eq. (7.31) yields exact results for polynomials of order at most equal to $2N-1$.Once again, the presence of the term $\frac{t_f - t_0}{2}$ is a consequence of the mapping between pseudospectral and physical time domains described in the relationships (7.19) and (7.20). The weights w_i can be computed with Eqs. (7.32), and (7.33).

$$w = flip(\tilde{w}) \tag{7.32}$$

$$\tilde{w}_i = \begin{cases} \frac{1}{(1-\tau_j)^2 \dot{P}^1_{N-1}},\ j \in [1, ...N] \\ \frac{2}{N},\ j = 0 \end{cases} \tag{7.33}$$

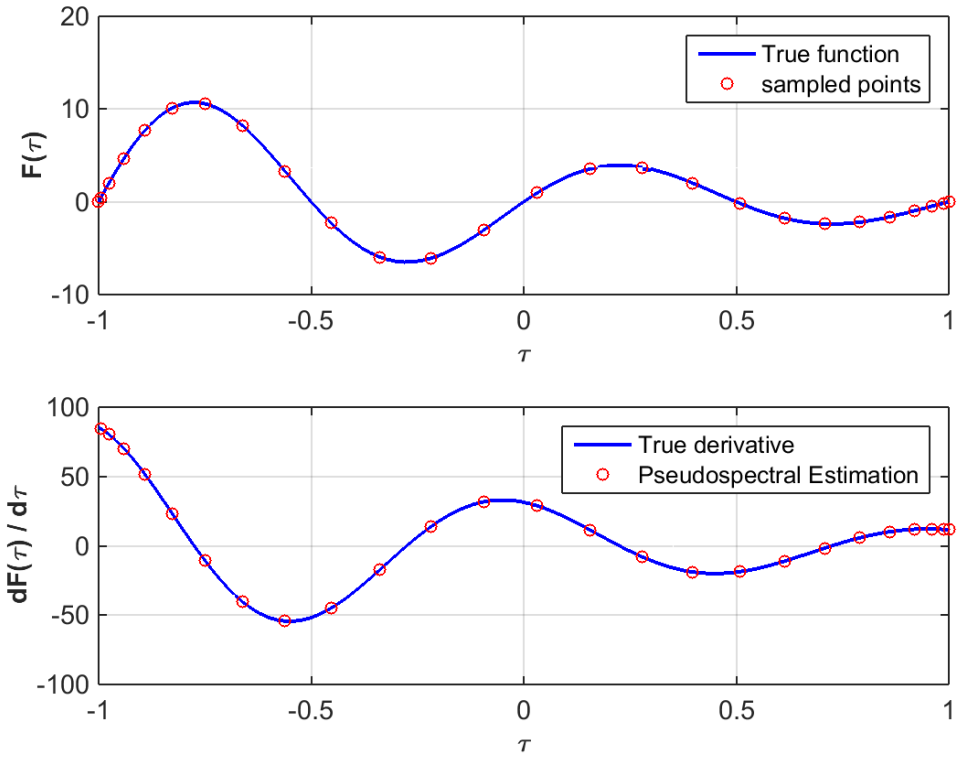


Figure 7.7: Test function $F(\tau) = Ae^{-\tau}\sin(\omega\tau)$ and its derivative estimation using 25 collocation nodes.

The operator *flip* simply multiplies the input by a factor equal to -1, and sorts the results in increasing order. An example is given in Fig. 7.8, where the test function $F(\tau)$

$$F(\tau) = 2\tau + 2 - \tau^2 \tag{7.34}$$

can be used to evaluate the integral.

Results are then compared with the analytical integral, and with the trapezoidal rule. Numerically, we get exactly the analytical result, that is 3.3333, while the application of the trapezoidal rule gives 3.3298. Once that a good approximation of the differential and integral operators have been described, we are ready to formulate the NLP problem which approximates the original OCP.

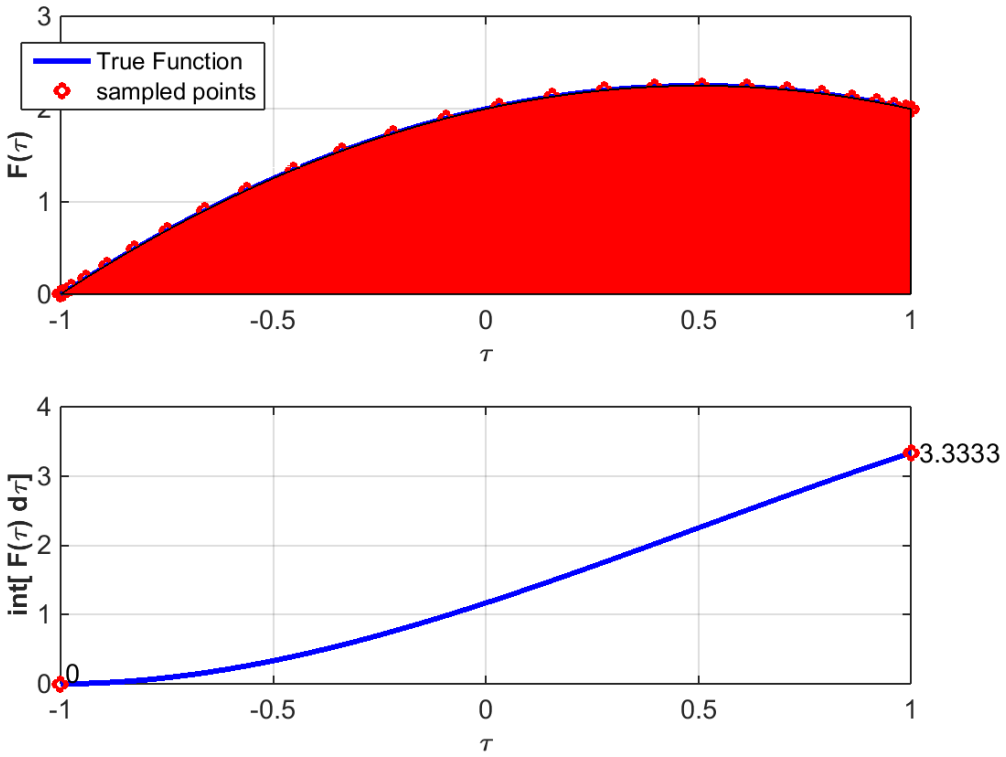


Figure 7.8: Test function $F(\tau) = 2\tau + 2 - \tau^2$ and its integral estimation using 25 collocation nodes.

7.2.6 Nonlinear Programming Problem

Minimize (maximize) the cost function J, for N nodes, $i = 1, \ldots, N$:

$$J = \Psi\left[\mathbf{X}_f\right] + \frac{t_f - t_0}{2} \sum_{i=0}^{N} w_i \Phi\left[\mathbf{X}_i, \mathbf{U}_i(\tau)\right] \tag{7.35}$$

subject to the nonlinear algebraic constraints

$$\mathbf{F} = \mathbf{D}\mathbf{X} - \frac{t_f - t_0}{2}\mathbf{f}(t, \mathbf{X}, \mathbf{U}) = \mathbf{0} \tag{7.36}$$

and to the path constraints

$$\mathbf{g}_L \leq \mathbf{G}\left(\mathbf{X}_i, \mathbf{U}_i\right) \leq \mathbf{g}_U \tag{7.37}$$

The discrete states and the controls are bounded, as in the continuous formulation.

$$\mathbf{x}_L \leq \mathbf{X}_i \leq \mathbf{x}_U \tag{7.38}$$

$$\mathbf{u}_L \leq \mathbf{U}_i \leq \mathbf{u}_U \tag{7.39}$$

This is the formal definition of the Nonlinear Programming Problem to solve. However, even if theoretically the problem could be solved, in practice further issues arise. In particular, the numerical conditioning of the problem, the exploitation of the Jacobian Matrix, and its computation play a major role in the quality of the results and the speed of the computation. This aspect is explored in the next Section.

7.3 Scaling procedure of the NLP problem

When the NLP problems are implemented, it can happen that they are ill-conditioned, that is, their numerical solution may be not accurate enough. This happens, for instance, if the variables involved in the problem are spread over several orders of magnitude. Therefore, for problems such as the entry guidance, where altitude, velocity, and flight-path angles significantly differ from each other in terms of magnitude, it is necessary to properly scale the NLP before solving it. But how can one establish if a NLP is well- or ill-conditioned?

A measure of the quality of a scaling method is the condition number of the Jacobian of the NLP defined in Eqs. (7.35)-(7.39), which in the general case is a rectangular matrix given by

$$\mathbf{Jac} = \begin{bmatrix} \nabla J \\ \nabla \mathbf{F} \\ \nabla \mathbf{G} \end{bmatrix} \tag{7.40}$$

where the operator ∇ is defined as

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial \mathbf{X}_0} & \frac{\partial}{\partial \mathbf{X}_1} & \frac{\partial}{\partial \mathbf{U}_1} & \cdots & \frac{\partial}{\partial \mathbf{X}_N} & \frac{\partial}{\partial \mathbf{U}_N} \end{bmatrix} \tag{7.41}$$

and the condition number is defined as the ratio between the maximum and the minimum eigenvalue of the Jacobian matrix.

$$C.N. = \frac{\max[eig(\mathbf{Jac})]}{\min[eig(\mathbf{Jac})]} \tag{7.42}$$

The closer to 1 the $C.N.$ is, the better-conditioned the matrix we are dealing with is. Specifically, we apply this metric to the Jacobian of the NLP to be solved because this matrix defines the search direction during the iterative process, and therefore a well-conditioned Jacobian is essential for solving the NLP defined in Eqs. (7.35)-(7.39) without excessive rounding errors. The problem moves thus towards the methods to properly scale the Jacobian matrix. This is what will be done in this section.

In detail, in the following sections, we will show how the Jacobian of the differential equations $\mathbf{F}(\mathbf{X})$ and the path constraints $\mathbf{G}(\mathbf{X})$ can be treated with different techniques. Specifically, three linear techniques, and two nonlinear techniques for the automatic scaling of the NLP previously defined will be introduced. They are:

- IsoScaling (IS) - Linear technique;
- Jacobian Rows Normalization (JRN) - Linear technique;
- Projected Jacobian Rows Normalization (PJRN) - Linear technique;
- Logarithmic (LOG) - Nonlinear technique;
- Inverse-Power (IP) - Nonlinear scaling;

While the first two methods are heritage of literature, the last three have been developed for SHEFEX-3, and their comparison with the existing methods suggest they can be considered a valid choice for several NLP problems.

7.3.1 Scaling of NLP States

The states $\mathbf{X}$ of the NLP problem are scaled using the standard linear transformation[58], regardless of the NLP scaling method (IS, JRN, etc.) that we use. Specifically, the scaled state $\tilde{\mathbf{X}}$ is given by:

$$\tilde{\mathbf{X}} = \mathbf{K_x} \cdot \mathbf{X} + \mathbf{b_x} \tag{7.43}$$

where $\mathbf{K_x}$ is a diagonal matrix, and $\mathbf{b_x}$ is a vector having the same dimensions as $\mathbf{X}$.

Since we always deal with bounded states, the elements of $\mathbf{K_x}$ and $\mathbf{b_x}$ are defined to be

$$\mathbf{K_{x_{ii}}} = \frac{1}{\mathbf{X_{U_i}} - \mathbf{X_{L_i}}}, \quad \mathbf{b_{x_i}} = -\frac{\mathbf{X_{L_i}}}{\mathbf{X_{U_i}} - \mathbf{X_{L_i}}} \tag{7.44}$$

Note that the transformation in Eq. (7.43) yields scaled states $\tilde{\mathbf{X}}$, which always lie in the interval $[0, 1]$. In case of unbounded states, artificial upper and lower boundaries are usually introduced[58].

7.3.2 Linear Techniques

Linear scaling techniques use a scaling of the form

$$\tilde{\mathbf{F}} = \mathbf{K_f} \cdot \mathbf{F}, \quad \tilde{\mathbf{G}} = \mathbf{K_g} \cdot \mathbf{G} \tag{7.45}$$

where $\mathbf{K_f}$ and $\mathbf{K_g}$ are diagonal matrices. The isoscaling (IS) method is one such technique where the constraints $\mathbf{F}$ are scaled exactly like the states, that is,

$$\mathbf{K_f} = \mathbf{K_x},$$

and $\mathbf{K_x}$ is given by Eq. (7.44)[58,65]. Note that isoscaling does not help in scaling properly the constraints $\mathbf{G}$, as no information coming from them are used (these constraints do not appear in Eq. (7.43). A possible refinement of this approach has been suggested by Rao[66], who uses randomly sampled points around the vector $\mathbf{X}$, and computes the mean of the norms of the Jacobian rows instead of the simple norm of the Jacobian rows evaluated at the current states and controls. Unfortunately, this technique may significantly increase the CPU time needed to compute the scaling coefficients, since

the Jacobian matrix must be evaluated many more times. Next, we introduce a simple linear scaling technique, which does not require additional Jacobian evaluations, and hence is much less computationally expensive.

7.3.2.1 Projected Jacobian Rows Normalization

Isoscaling bases the scaling of the constraints solely on the scaling of the states. In other words, it does not take into account the relationship between the states and the constraints, which is represented in linearized form by the Jacobian. Conversely, Jacobian Rows Normalization (JRN) only considers this relationship, without involving the states' normalization in the process. Specifically, in the JRN technique, the diagonal elements of $\mathbf{K_f}$ and $\mathbf{K_g}$ are given by:

$$\mathbf{K_{f_{ii}}} = \frac{1}{|\nabla \mathbf{F}|_i}, \quad \mathbf{K_{g_{ii}}} = \frac{1}{|\nabla \mathbf{G}|_i} \tag{7.46}$$

The projected Jacobian rows normalization (PJRN) technique, which we propose considers both of these factors. Specifically, in PJRN, the diagonal elements of $\mathbf{K_f}$ and $\mathbf{K_g}$ are given by:

$$\mathbf{K_{f_{ii}}} = \frac{1}{\left|\nabla \mathbf{F} \cdot \mathbf{K_x^{-1}}\right|_i}, \quad \mathbf{K_{g_{ii}}} = \frac{1}{\left|\nabla \mathbf{G} \cdot \mathbf{K_x^{-1}}\right|_i} \tag{7.47}$$

As we will show in the summary of the tests, this scaling generally leads to a better-conditioned Jacobian matrix, and to more uniformly distributed singular values.

The Jacobian of the PJRN-scaled NLP is given by:

$$\tilde{\mathbf{Jac}} = \begin{bmatrix} \tilde{\nabla}\tilde{J} \\ \tilde{\nabla}\tilde{\mathbf{F}} \\ \tilde{\nabla}\tilde{\mathbf{G}} \end{bmatrix} = \begin{bmatrix} K_J \cdot \nabla J \cdot \mathbf{K_x^{-1}} \\ \mathbf{K_F} \cdot \nabla \mathbf{F} \cdot \mathbf{K_x^{-1}} \\ \mathbf{K_G} \cdot \nabla \mathbf{G} \cdot \mathbf{K_x^{-1}} \end{bmatrix} \tag{7.48}$$

where the operator $\tilde{\nabla}$ is defined as

$$\tilde{\nabla} = \begin{bmatrix} \frac{\partial}{\partial \tilde{\mathbf{X}}_\mathbf{0}} & \frac{\partial}{\partial \tilde{\mathbf{X}}_\mathbf{1}} & \frac{\partial}{\partial \tilde{\mathbf{U}}_\mathbf{1}} \cdots & \frac{\partial}{\partial \tilde{\mathbf{X}}_\mathbf{N}} & \frac{\partial}{\partial \tilde{\mathbf{U}}_\mathbf{N}} \end{bmatrix} \tag{7.49}$$

and K_J is a parameter which normalizes the cost function J. $\mathbf{K_x}$ is given by Eq. (7.44), and $\mathbf{K_f}$ and $\mathbf{K_g}$ are given by Eq. (7.46).

7.3.3 Nonlinear Techniques

Nonlinear scaling techniques generalize the second relationship given by Eq. (7.48). Note that it is also possible to scale $\mathbf{F}$ with a nonlinear scaling technique, however, we will take advantage of the boundedness of $\mathbf{G}$, and hence we only consider the nonlinear scaling of $\mathbf{G}$ in this work. Specifically, we propose the logarithmic and the inverse-power scaling techniques.

7.3.3.1 Logarithmic Scaling

The first nonlinear scaling technique we propose is the one based on the natural logarithm, in which case the scaled constraint function is given by:

$$\tilde{\mathbf{G}} = \log\left(\mathbf{G} + \mathbf{C}\right) \tag{7.50}$$

The constant vector $\mathbf{C}$ ensures that the argument of the logarithm is always greater than or equal to 1. Specifically, since the constraint function is bounded from below by $\mathbf{g_L}$, according to Eq. (7.37), we choose $\mathbf{C}$ to be given by

$$\mathbf{C} = -\mathbf{g_L} + \mathbf{1} \tag{7.51}$$

When the constraints are intrinsically positive (e.g., when we consider the dynamic pressure or the heat rate, as in the case of the Space-Shuttle entry problem), this simply reduces to

$$\mathbf{C} = \mathbf{1} \tag{7.52}$$

where $\mathbf{1}$ is a vector of ones with the same dimensions as $\mathbf{G}$.

The Jacobian of a logarithmically scaled NLP is of the form reported in Eq. (7.48), where $\nabla\tilde{\mathbf{G}}$ is given by

$$\tilde{\nabla}\tilde{\mathbf{G}} = \frac{\mathbf{1}}{\mathbf{G} + \mathbf{C}} \cdot \nabla\mathbf{G} \cdot \mathbf{K_x^{-1}} \tag{7.53}$$

Note that since the logarithmic scaling only affects $\mathbf{G}$, $\mathbf{K_f}$ must be chosen using a linear scaling technique such as IS, JRN, or PJRN.

7.3.3.2 Inverse-Power Scaling

The second nonlinear scaling technique we propose is the inverse-power scaling technique, in which case the scaled constraint function is given by

$$\tilde{\mathbf{G}} = \left(\mathbf{G} + \mathbf{C}\right)^{\frac{\mathbf{1}}{\mathbf{n}}} \tag{7.54}$$

where $\mathbf{n}$ is a vector, positive integer, and $\mathbf{C}$ is chosen so that $\mathbf{G} + \mathbf{C}$ is always greater than or equal to 1. Specifically, in the present work, we always chose $\mathbf{C}$ according to Eq. (7.51). Furthermore, we chose $\mathbf{n}$ to be given by

$$\mathbf{n} = ceil\left[log(\mathbf{G} + \mathbf{C})\right] \tag{7.55}$$

Equations (7.54) and (7.55) are meant element-wise. Like the logarithmic scaling, the inverse-power scaling is only applied to the constraint function $\mathbf{G}$. Hence the Jacobian of an inverse-power scaled NLP is of the form of Eq. (7.48), where $\nabla\tilde{\mathbf{G}}$ is given by

$$\tilde{\nabla}\tilde{\mathbf{G}} = \frac{\mathbf{1}}{\mathbf{n}\left(\mathbf{G} + \mathbf{C}\right)^{\frac{\mathbf{n-1}}{\mathbf{n}}}} \cdot \nabla\mathbf{G} \cdot \mathbf{K_x^{-1}} \tag{7.56}$$

Note that since the inverse power scaling only affects $\mathbf{G}$, $\mathbf{K_f}$ must be chosen using a linear scaling technique such as IS, JRN, or PJRN.

7.3.4 Test Problems Definition

The aforementioned scaling techniques have been tested for the Space-Shuttle entry problem (SS), the reorientation of an asymmetric body (AB), and the range maximization of a hang glider (HG). A complete description of the problems can be found in[58]. All of the problems have variables and constraints defined over a wide range of magnitudes. Furthermore, constraints on the maximum heat rate, the maximum angular speed and the maximum load factor are included.

For testing the routine, the scaling techniques have been applied using a set of 50 nodes with two different NLP solvers, SNOPT and IPOPT, based on Sequential Quadratic Programming and Interior-Point methods, respectively. The dynamics $\mathbf{F}$ and constraints $\mathbf{G}$ have been scaled using the three linear techniques (IS, JRN, and PJRN), while the nonlinear techniques have only been used on the constraints. The optimality and feasibility tolerances are set to 10^{-15}. Moreover, for each problem, the results obtained without scaling (NONE) and with solvers' internal scaling procedures (AUTO) are reported. Tables 7.1 and 7.2 show the ratio of the condition numbers of the Jacobian matrices of the non-scaled and scaled NLPs, that is,

$$C.N.Ratio = \frac{cond\left[\mathbf{Jac}(\mathbf{X})\right]}{cond\left[\tilde{\mathbf{Jac}}(\tilde{\mathbf{X}})\right]} \tag{7.57}$$

Note that a larger C.N. ratio is better, since it implies a larger improvement in the Jacobian conditioning. In addition to the C.N.Ratios, Tables 7.1 and 7.2 show the CPU times required to compute the scaling matrices, the CPU times required to solve the scaled NLPs using SNOPT and IPOPT, and the cost function J obtained by solving the several forms of the NLP problems. Best results obtained for each of the problem have been pointed out. All of the examples have been solved with a cold start, that is, no helpful initial guesses were provided.

For all the cases we have considered, the proposed scaling techniques improve the CPU time required to solve the NLP problem. For instance, when SNOPT is used, for the Space-Shuttle problem the use of the PJRN for the dynamics, in combination with the JRN for the constraint, allows to speed up the process up to 600% w.r.t. the traditional combination IS-JRN. The C.N. Ratio for the same problem can be up to $3.37 \cdot 10^7$, and improves the Jacobian conditioning more than the traditional linear methodologies.

For some of the techniques we proposed it is also possible to observe slight improvements in the cost functions (for instance in the case of the AB problem, when the PJRN is employed for both the dynamics and the constraints in combination with IPOPT). These improvements hold for the Space Shuttle and the reorientation problem. For the hang-glider problem, no variations in the cost functions were observed, but the CPU time is positively affected by the use of the proposed techniques in several

cases. When no scaling or NLP solvers' scaling routines were used, for the Space Shuttle problem it was not even possible to get a valid solution. For the other problems, valid solutions were obtained, but with worse performances. For the purpose of SHEFEX-3, the Projected Jacobian Rows Normalization technique will be used, as it provides the best performance in terms of C.N. Ratio.

Table 7.1: Automatic Scaling Performances - SNOPT.

	Technique		Scaling Performances		Performances	
Problem	$\nabla\tilde{\mathbf{F}}$	$\nabla\tilde{\mathbf{G}}$	C.N. Ratio	CPU Time (s)	J	CPU Time (s)
SS	IS	JRN	5.857990e+006	3.025e-001	-3.063849e+001	5.795e+000
	IS	PJRN	9.677610e+005	4.550e-002	-3.063850e+001	3.165e+000
	IS	LOG	5.823251e+006	4.619e-002	-3.063849e+001	4.596e+000
	IS	IP	5.881658e+006	3.647e-002	-3.063849e+001	4.598e+000
	JRN	JRN	7.654331e+005	4.447e-001	-3.063850e+001	2.621e+000
	JRN	PJRN	7.635124e+005	1.761e-001	-3.063850e+001	1.898e+000
	JRN	LOG	7.654697e+005	1.987e-001	-3.063850e+001	2.124e+000
	JRN	IP	7.654017e+005	1.624e-001	-3.063850e+001	1.513e+000
	PJRN	JRN	3.063962e+006	1.313e-001	-3.063850e+001	1.083e+000
	PJRN	PJRN	3.370484e+007	2.972e-002	-3.063849e+001	1.923e+000
	PJRN	LOG	5.172896e+006	3.178e-002	-3.063848e+001	2.398e+000
	PJRN	IP	9.801066e+006	2.967e-002	-3.063850e+001	1.482e+000
	NONE	NONE	-	-	solution not valid	-
	AUTO	AUTO	-	-	solution not valid	-
AB	IS	JRN	5.990499e+003	2.427e-001	3.069326e+001	3.457e+000
	IS	PJRN	1.389897e+005	3.509e-002	3.013058e+001	2.678e+000
	IS	LOG	1.002052e+002	4.573e-002	2.999047e+001	2.455e+000
	IS	IP	3.340174e+001	6.262e-002	2.999047e+001	3.694e+000
	JRN	JRN	5.431878e+003	3.0844-001	3.013108e+001	3.802e+000
	JRN	PJRN	3.139037e+004	1.922e-001	3.013106e+001	3.220e+000
	JRN	LOG	1.152042e+002	1.783e-001	3.013108e+001	3.178e+000
	JRN	IP	3.840144e+001	1.590e-001	3.013107e+001	2.758e+000
	PJRN	JRN	4.420080e+003	1.817e-001	2.999047e+001	3.979e+000
	PJRN	PJRN	1.131537e+005	3.251e-002	3.013107e+001	2.995e+000
	PJRN	LOG	1.152033e+002	4.600e-002	3.071294e+001	2.827e+000
	PJRN	IP	3.840110e+001	3.861e-002	3.070010e+001	1.775e+000
	NONE	NONE	-	-	3.013108e+001	3.733e+000
	AUTO	AUTO	-	-	3.013108e+001	2.460e+000
HG	IS	JRN	3.53921e+001	1.006e-001	-1.248032e+003	1.041e+000
	IS	PJRN	3.281319e+001	1.528e-002	-1.248032e+003	8.580e-001
	IS	LOG	1.854903e+001	9.602e-003	-1.248032e+003	5.453e-001
	IS	IP	1.071138e+001	1.277e-002	-1.248032e+003	5.929e-001
	JRN	JRN	9.384483e+000	1.285e-001	-1.248032e+003	7.432e-001
	JRN	PJRN	9.024341e+000	5.274e-002	-1.248032e+003	6.736e-001
	JRN	LOG	6.165254e+000	5.309e-002	-1.248032e+003	5.719e-001
	JRN	IP	3.848432e+000	5.291e-002	-1.248032e+003	5.770e-001
	PJRN	JRN	8.954468e+003	4.344e-002	-1.248032e+003	4.417e-001
	PJRN	PJRN	8.657115e+003	1.431e-002	-1.248032e+003	6.012e-001
	PJRN	LOG	6.061327e+003	1.068e-002	-1.248032e+003	4.340e-001
	PJRN	IP	3.835293e+003	1.005e-002	-1.248032e+003	4.355e-001
	NONE	NONE	-	-	-1.248032e+003	2.346e+000
	AUTO	AUTO	-	-	-1.248032e+003	2.770e+000

Table 7.2: Automatic Scaling Performances - IPOPT.

	Technique		Scaling Performances		Performances	
Problem	$\nabla\tilde{\mathbf{F}}$	$\nabla\tilde{\mathbf{G}}$	C.N. Ratio	CPU Time (s)	J	CPU Time (s)
SS	IS	JRN	5.857990e+006	3.025e-001	-3.063850e+001	6.116e+001
	IS	PJRN	9.677610e+005	4.550e-002	-3.063850e+001	6.020e+001
	IS	LOG	5.823251e+006	4.619e-002	-3.063850e+001	4.424e+001
	IS	IP	5.881658e+006	3.647e-002	-3.063850e+001	2.659e+001
	JRN	JRN	7.654331e+005	4.447e-001	-3.063849e+001	1.833e+001
	JRN	PJRN	7.635124e+005	1.761e-001	-3.063849e+001	3.173e+001
	JRN	LOG	7.654697e+005	1.987e-001	-3.063850e+001	4.804e+001
	JRN	IP	7.654017e+005	1.624e-001	-3.063849e+001	1.787e+001
	PJRN	JRN	3.063962e+006	1.313e-001	-3.063849e+001	1.917e+001
	PJRN	PJRN	3.370484e+007	2.972e-002	-3.063849e+001	1.758e+002
	PJRN	LOG	5.172896e+006	3.178e-002	-3.063850e+001	4.544e+001
	PJRN	IP	9.801066e+006	2.967e-002	-3.063850e+001	7.329e+001
	NONE	NONE	-	-	solution not valid	-
	AUTO	AUTO	-	-	solution not valid	-
AB	IS	JRN	5.990499e+003	2.427e-001	2.999047e+001	5.186e+000
	IS	PJRN	1.389897e+005	3.509e-002	3.013107e+001	6.322e+000
	IS	LOG	1.002052e+002	4.573e-002	2.999047e+001	2.475e+000
	IS	IP	3.340174e+001	6.262e-002	2.999047e+001	4.782e+000
	JRN	JRN	5.431878e+003	3.0844-001	3.013107e+001	5.568e+000
	JRN	PJRN	3.139037e+004	1.922e-001	3.013107e+001	9.661e+001
	JRN	LOG	1.152042e+002	1.783e-001	3.069973e+001	2.720e+000
	JRN	IP	3.840144e+001	1.590e-001	2.999047e+001	6.105e+000
	PJRN	JRN	4.420080e+003	1.817e-001	2.999047e+001	3.713e+000
	PJRN	PJRN	1.131537e+005	3.251e-002	3.013107e+001	7.307e+001
	PJRN	LOG	1.152033e+002	4.600e-002	3.069973e+001	2.419e+000
	PJRN	IP	3.840110e+001	3.861e-002	3.013107e+001	5.895e+000
	NONE	NONE	-	-	2.999047e+001	2.059e+001
	AUTO	AUTO	-	-	2.999047e+001	1.733e+001
HG	IS	JRN	3.53921e+001	1.006e-001	-1.248032e+003	2.239e+000
	IS	PJRN	3.281319e+001	1.528e-002	-1.248032e+003	1.437e+000
	IS	LOG	1.854903e+001	9.602e-003	-1.248032e+003	1.467e+000
	IS	IP	1.071138e+001	1.277e-002	-1.248032e+003	2.045e+000
	JRN	JRN	9.384483e+000	1.285e-001	-1.248032e+003	1.370e+000
	JRN	PJRN	9.024341e+000	5.274e-002	-1.248032e+003	1.439e+000
	JRN	LOG	6.165254e+000	5.309e-002	-1.248032e+003	1.507e+000
	JRN	IP	3.848432e+000	5.291e-002	-1.248032e+003	1.363e+000
	PJRN	JRN	8.954468e+003	4.344e-002	-1.248032e+003	1.825e+000
	PJRN	PJRN	8.657115e+003	1.431e-002	-1.248032e+003	1.517e+000
	PJRN	LOG	6.061327e+003	1.068e-002	-1.248032e+003	1.777e+000
	PJRN	IP	3.835293e+003	1.005e-002	-1.248032e+003	1.657e+000
	NONE	NONE	-	-	-1.248032e+003	3.463e+001
	AUTO	AUTO	-	-	-1.248032e+003	8.848e+001

7.4 Computation of the Jacobian

Another important step of the transcription is the computation of the Jacobian associated with the NLP. In SPARTAN the Jacobian matrix is exploited to take advantage of the

sparsity patterns associated with the use of pseudospectral methods. In this section the computation of the Jacobian matrix is described.

7.4.1 Analysis of Continuous System

Experience shows that, while for simple systems a more detailed analysis of the Jacobian can be avoided, in complex problems like atmospheric reentry a solid knowledge of its structure is very helpful and significantly increases the speed of computation and in some cases the quality of the results, as the Jacobian matrix is a direct representation of the dynamics of the system. Moreover, it gives relevant information about the cost function evolution (which determines the direction of search during the solution generation of the NLP), and the constraints. It is then convenient to look at the continuous functions representing the specific problem. The first step is to perform a systematic analysis of the functions, which define the NLP to be solved. The idea is to create a set of perturbation states and controls, and evaluate the cost function, the dynamics of the system and the constraints.

To do this, random values of the states and controls are generated using the initial guesses provided to the NLP solver. The rationale for this is that randomly-generated data can tell us about the dependencies of the functions w.r.t. the states and the controls. This information will be used to exploit the sparsity of the Jacobian. Indeed, when no dependency of a function from a variable is recognized, the corresponding Jacobian element is equal to 0. Moreover, assuming that no a-priori knowledge of the solution is known, an acceptable initial guess along the trajectory can be computed as linear interpolation between the initial and final values $(\mathbf{X}_0, \mathbf{U}_0)$ and $(\mathbf{X}_f, \mathbf{U}_f)$. Let us define their mid-values as follows.

$$\mathbf{X}_m = \frac{\mathbf{X}_0 + \mathbf{X}_f}{2} \tag{7.58}$$

$$\mathbf{U}_m = \frac{\mathbf{U}_0 + \mathbf{U}_f}{2} \tag{7.59}$$

Assuming that the states and the controls are bounded, the random values of the variables needed to evaluate the functions can simply be computed according to normal distributions around these mean values,

$$\mathbf{X}_p = N(\mathbf{X}_m, \sigma_x^2) \tag{7.60}$$

$$\mathbf{U}_p = N(\mathbf{U}_m, \sigma_u^2) \tag{7.61}$$

where σ_x and σ_u are the standard deviations assumed to be proportional to the difference between the upper and lower boundaries of the variables. In this way, a certain number of perturbation vectors can be generated.

It is important to stress that here we are only interested in generating the output from the continuous functions, so it is not important in this phase to have real state and control data (i.e., data which satisfy our optimal-control problem). The functions

are evaluated using the inputs expressed by Eqs. (7.60) and (7.61). The objective is to generate the dependency matrices related to the cost function, the dynamics and the constraints. These matrices will have a column for each classical continuous variable, (i.e., time, states and controls), and a row for each constraining function. The elements of the matrices are defined equal to 0 when no direct dependency is recognized and 1 when a direct dependency is recognized. For example, assuming a system with two

Table 7.3: Definition of generic dependency matrix.

	variable#1	**variable**#2	$\cdots$	**variable**#M
function#1	0 (no dependency)	1 (dependency)	$\cdots$	1
function#2	1 (dependency)	0 (dependency)	$\cdots$	0
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
function#N	1 (dependency)	1 (dependency)	$\cdots$	0

states and one control channel,

$$\begin{array}{rcl} \dot{x}_1(t) & = & 2x_1^3 + 3x_2 + u_1^2 \\ \dot{x}_2(t) & = & 5\sin x_1 - 3u_1 \end{array} \tag{7.62}$$

the function describing the dynamics will have the 2×4 dependency matrix reported in Table 7.4.

Table 7.4: Dynamics dependency matrix for 2-states system.

	t	x_1	x_2	u_1
$\dot{x}_1$	0	1	1	1
$\dot{x}_2$	0	1	0	1

A significant example, used for the development of the tool, is the Space-Shuttle reentry problem (as treated by Betts[58]). In this problem the aim is to maximize the final latitude ϕ

$$J = \max(\phi_F) \tag{7.63}$$

of the vehicle, subject to the following differential equations

$$
\begin{aligned}
\dot{h} &= V\ \sin\gamma \\
\dot{\theta} &= \frac{V\ \sin\psi\ \cos\gamma}{r\ \cos\phi} \\
\dot{\phi} &= \frac{V}{r}\ \cos\gamma\ \cos\psi \\
\dot{V} &= -\frac{D}{m} - g\ \sin\gamma \\
\dot{\gamma} &= \frac{L}{m\,V}\cos\sigma + \cos\gamma\left(\frac{V}{r} - \frac{g}{V}\right) \\
\dot{\psi} &= \frac{L}{m\,V\ \cos\gamma}\ \sin\sigma + \frac{V}{r\cos\phi}\cos\gamma\ \sin\psi\ \sin\phi
\end{aligned}
\tag{7.64}
$$

with the aerodynamic forces computed according to

$$
\begin{aligned}
L &= \frac{1}{2}\rho V^2 S C_L(\alpha) \\
D &= \frac{1}{2}\rho V^2 S C_D(\alpha)
\end{aligned}
\tag{7.65}
$$

For this case, the states are the altitude h, the longitude θ, the latitude ϕ, the velocity modulus V, the flight-path angle γ, and the flight-path heading ψ, while the controls are the angle of attack α and the bank angle σ. A constraint, limiting the heat flux, is included.

$$
\dot{Q} = k_Q(c_0 + c_1\alpha + c_2\alpha^2)\sqrt{\rho}(c_3 V^{3.07}) <= \dot{Q}_U \tag{7.66}
$$

In a similar fashion to what has been done for the system described by Eq. (7.62), we can derive the dependency matrices. The dependency matrix for the cost function has dimensions 2×9. The Mayer and Lagrange contributions are treated in a separate way, as the Mayer term involves only the last columns of the Jacobian, while the Lagrange term involves potentially the entire row of the Jacobian associated with the cost function J. The dynamics dependency Matrix has, for the analysed problem,

Table 7.5: Cost function dependency matrix for Space Shuttle reentry case.

	t	h	θ	ϕ	V	γ	ψ	α	σ
J_{MAYER}	0	0	0	1	0	0	0	0	0
$J_{LAGRANGE}$	0	0	0	0	0	0	0	0	0

dimensions 6×9. For the heat flux, we can write a corresponding 1×9 dependency matrix. This information can be rearranged in the matrices defined by Eqs. (7.67)-(7.78), which will be used to compute the Jacobian in an efficient way, as it will be

Table 7.6: Dynamics dependency matrix for the Space Shuttle reentry case.

	t	h	θ	ϕ	V	γ	ψ	α	σ
$\dot{h}$	0	0	0	1	1	0	0	0	0
$\dot{\theta}$	0	1	0	1	1	1	1	0	0
$\dot{\phi}$	0	1	0	0	1	1	1	0	0
$\dot{V}$	0	1	0	0	1	1	0	1	0
$\dot{\gamma}$	0	1	0	0	1	1	0	1	1
$\dot{\psi}$	0	1	0	1	1	1	1	1	1

Table 7.7: Constraint dependency matrix for Space Shuttle reentry case.

	t	h	θ	ϕ	V	γ	ψ	α	σ
$\dot{Q}$	0	1	0	0	1	0	0	1	0

shown in the next section.

$$M_{cost_{\Phi,t}} = 0 \tag{7.67}$$

$$M_{cost_{\Psi,t}} = 0 \tag{7.68}$$

$$\mathbf{M}_{cost_{\Phi,x}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{7.69}$$

$$\mathbf{M}_{cost_{\Psi,x}} = \mathbf{0}_{1\times 6} \tag{7.70}$$

$$\mathbf{M}_{cost_{\Phi,u}} = \mathbf{0}_{1\times 2} \tag{7.71}$$

$$\mathbf{M}_{cost_{\Psi,u}} = \mathbf{0}_{1\times 2} \tag{7.72}$$

$$\mathbf{M}_{dyn,t} = \mathbf{0}_{1\times 6} \tag{7.73}$$

$$\mathbf{M}_{dyn,x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{7.74}$$

$$\mathbf{M}_{dyn,u} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{7.75}$$

$$M_{cons,t} = 0 \tag{7.76}$$

$$\mathbf{M}_{cons,x} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{7.77}$$

$$\mathbf{M}_{cons,u} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{7.78}$$

7.4.2 Hybridization of Jacobian

Let us now consider the general structure of the Jacobian associated with the NLP problem deriving from the application of FRPM, and its specific application to the already cited case of the Space Shuttle entry problem.

7.4.2.1 General structure of Jacobian associated with the FRPM

In the most general case, considering n_s states, n_c controls, n_g constraints, N collocation points and unknown final time t_F, the Jacobian associated with the transcription of an autonomous system of equations (as in the examples treated here) will be expressed as a matrix having the following dimensions and structure.

$$\dim(\mathbf{Jac}) = [N \cdot (n_s + n_g) + 1] \times [(N+1) \cdot n_s + N \cdot n_c + 1] \tag{7.79}$$

While the dimensions of the Jacobian do not vary, once the variables and the constraints are given, the kind of sparsity patterns, which will appear in it, depends on the ordering of the NLP state vector. To take advantage of the information provided by the dependency matrices, and to maintain a consistency between the states and the controls associated to each node, the following order for the NLP variable is proposed:

$$\mathbf{X}_{NLP} = \left[\begin{array}{c|cc|cc|c|cc|c} \mathbf{X}_0 & \mathbf{X}_1 & \mathbf{U}_1 & \mathbf{X}_2 & \mathbf{U}_2 & .. & .. & \mathbf{X}_N & \mathbf{U}_N & t_F \end{array} \right]^T \tag{7.80}$$

We can observe how the initial control $\mathbf{U}_0$ does not appear in Eq. (7.80). This is due to the choice of the FRPM as transcription method instead of the traditional RPM. The initial control can be extrapolated once the NLP is solved. Since the Jacobian is by definition the matrix representing the partial derivatives of a given set of functions (i.e., our NLP constraints) w.r.t. their variables, this set and its order must be defined. For the NLP, they are all the constraints defined during the transcription of the problem, that is, the cost function J, the dynamics $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N\}$, and, when defined, the constraints $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_N\}$. All together they are the constraints $\mathbf{C}(\mathbf{X}_{NLP})$ of the NLP:

$$\mathbf{C}(\mathbf{X}_{NLP}) = \left[\begin{array}{c|ccc|ccc} J & \mathbf{f}_1 & \mathbf{f}_2 & ... & \mathbf{f}_N & \mathbf{g}_1 & \mathbf{g}_2 & ... & \mathbf{g}_N \end{array} \right]^T \tag{7.81}$$

The Jacobian derived from these definitions is

$$\mathbf{Jac} = \left[\frac{\partial \mathbf{C}}{\partial \mathbf{X}_{NLP}}\right] = \left[\begin{array}{c|c|c|c|c|c|c|c|c|c} \frac{\partial J}{\partial \mathbf{X}_0} & \frac{\partial J}{\partial \mathbf{X}_1} & \frac{\partial J}{\partial \mathbf{U}_1} & \frac{\partial J}{\partial \mathbf{X}_2} & \frac{\partial J}{\partial \mathbf{U}_2} & \cdots & \vdots & \frac{\partial J}{\partial \mathbf{X}_N} & \frac{\partial J}{\partial \mathbf{U}_N} & \frac{\partial J}{\partial t_F} \\ \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_2} & \cdots & \vdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_N} & \frac{\partial \mathbf{f}_1}{\partial t_F} \\ \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_2} & \cdots & \vdots & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_N} & \frac{\partial \mathbf{f}_2}{\partial t_F} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{U}_N} & \frac{\partial \mathbf{f}_N}{\partial t_F} \\ \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_N} & \frac{\partial \mathbf{g}_1}{\partial t_F} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \mathbf{g}_N}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial \mathbf{g}_N}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{U}_N} & \frac{\partial \mathbf{g}_N}{\partial t_F} \end{array}\right] \tag{7.82}$$

This Jacobian matrix can be computed numerically in different ways (e.g., with the classical finite differences scheme or using the complex-step derivative technique[64]). These are not the best approaches, since they do not consider the theoretical knowledge contained in the definition of the discrete operator $\mathbf{D}$, nor do take full advantage from the intrinsic sparsity associated with the use of pseudospectral methods. Therefore, let us express the Jacobian as the sum of three different contributions:

$$\mathbf{Jac} = \mathbf{Jac}_{PS} + \mathbf{Jac}_{Num} + \mathbf{Jac}_{Th} \tag{7.83}$$

We can now analyse each of these terms and show how to compute them.

7.4.2.2 Pseudospectral Jacobian

This part of the Jacobian matrix is intrinsically related to the use of the FRPM. More specifically, it can be seen as the contribution to the Jacobian and to the constraints represented into Eq. (7.82) given by the use of the discrete differential matrix $\mathbf{D}$. In the frame of the discretization of the dynamics, it represents the term

$$\mathbf{D} \cdot \mathbf{X} \tag{7.84}$$

From a pure algebraic point of view, the differential operator can be seen as a set of linear combinations of the nodal values of each of the states. The pseudospectral Jacobian $\mathbf{Jac}_{PS}$ is entirely defined once $\mathbf{D}$ is computed. More explicitly, the pseudospectral Jacobian matrix can be defined as follows

$$\mathbf{Jac}_{PS} = \left[\begin{array}{ccc|c} & \mathbf{0}_{1\times[(N+1)\cdot n_s+N\cdot n_c+1]} & & \\ \tilde{\mathbf{D}}_{\mathbf{1,0}} \cdots & \cdots & \tilde{\mathbf{D}}_{\mathbf{1,N}} & \\ \vdots \quad \vdots & \vdots & \vdots & \mathbf{0}_{[N\cdot(n_s+n_g)+1\times 1]} \\ \tilde{\mathbf{D}}_{\mathbf{N,0}} \cdots & \cdots & \tilde{\mathbf{D}}_{\mathbf{N,N}} & \\ & \mathbf{0}_{Nn_g\times[(N+1)\cdot n_s+N\cdot n_c+1]} & & \end{array}\right] \tag{7.85}$$

where

$$\tilde{\mathbf{D}}_{\mathbf{i,j}} = D_{i,j}\mathbf{I}_{n_s},\ j \in [0, N] \tag{7.86}$$

and I_{n_s} is the identity matrix of dimension n_s; the elements $D_{i,j}$ are defined by Eq. (7.29). The pseudospectral Jacobian is computed only once, before the beginning of the actual optimization process. Moreover, the accuracy of its computation is a consequence of how good the estimate of the roots of the Legendre-Radau polynomials is, and not of the errors given by the approximation due to the use of numerical differentiation techniques.

7.4.2.3 Numerical Jacobian

The numerical Jacobian can be computed using the classical finite differences or the complex-step derivative methods[64]. In particular, the second technique is very interesting as it removes the cancellation errors, which are a consequence of the use of the classical finite-differences scheme. This method is based on the hypothesis of having continuous functions, and consequently the functions defining the OCP can be seen as real parts of complex functions. Under these premises, the Cauchy-Riemann conditions for the complex function $u + iv = f(x + iy)$ are valid:

$$\frac{\partial u}{\partial x} = \frac{\partial y}{\partial v} \tag{7.87}$$

Then, by definition, the derivative of the real part of the function f can be approximated by

$$Re\left(\frac{df}{dx}\right) = \frac{\partial u}{\partial x} = \frac{\partial y}{\partial v} \cong \frac{Im\left(f(x + ih)\right)}{h} \tag{7.88}$$

where h is the numerical step used to perturb the functions, generally assumed to be very small. Alternatively, the dual-step approach[33,34] can be used. This choice yields to a slightly heavier computational load, but has the advantage to provide exact results.

The approach defined in Eq. (7.88) will be applied to all the elements of the Jacobian recognized by the dependency matrices. The chosen ordering for the state vector makes this operation quite intuitive, as the nonzero elements of the dependency matrices will represent the patterns of computation for the numerical derivatives. We can express this contribution as

$$\mathbf{Jac}_{Num} = \mathbf{Jac}_{Cont} \circ \mathbf{M}_{GD} \tag{7.89}$$

where $\mathbf{Jac}_{Cont}$ is the Jacobian considering only the continuous functions defining the

problem (i.e., considering the matrix $\mathbf{D}$ equal to 0), excluding the last column,

$$\mathbf{Jac}_{Cont} = \left[\frac{\partial \mathbf{C}}{\partial \mathbf{X}_{NLP}}\right]_{\mathbf{D}=\mathbf{0}} = -k_t \left[\begin{array}{cccccccccc|c} \frac{\partial J}{\partial \mathbf{X}_0} & \frac{\partial J}{\partial \mathbf{X}_1} & \frac{\partial J}{\partial \mathbf{U}_1} & \frac{\partial J}{\partial \mathbf{X}_2} & \frac{\partial J}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial J}{\partial \mathbf{X}_N} & \frac{\partial J}{\partial \mathbf{U}_N} & \\ \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{f}_1}{\partial \mathbf{U}_N} & \\ \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial \mathbf{f}_2}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{f}_2}{\partial \mathbf{U}_N} & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \mathbf{0}_{[N\cdot(n_s+n_g)+1\times 1]} \\ \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{f}_N}{\partial \mathbf{U}_N} & \\ \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_N} & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \frac{\partial \mathbf{g}_N}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{U}_1} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{X}_2} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{U}_2} & \cdots & \cdots & \frac{\partial \mathbf{g}_N}{\partial \mathbf{X}_N} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{U}_N} & \end{array}\right]_{\mathbf{D}=\mathbf{0}}$$

where k_t is $\frac{t_f - t_0}{2}$ for the derivatives of elements $\mathbf{f}_i$, and equal to 1 for all other elements. $\mathbf{M}_{GD}$ is a matrix having the same dimensions of the Jacobian, and defined by Eq. (7.90),

$$\mathbf{M}_{GD} = \left[\begin{array}{ccc} OR(\mathbf{M}_{cost_{\Phi,x}}, \mathbf{M}_{cost_{\Psi,x}}) & \mathbf{M}_{cost_{\Psi,x}} & \mathbf{M}_{cost_{\Phi,x}} \cdots \\ \mathbf{M}_{dyn,x} & \mathbf{M}_{dyn,x} & \mathbf{M}_{dyn,u} \cdots \\ \vdots & \vdots & \vdots \\ \mathbf{M}_{dyn,x} & \mathbf{M}_{dyn,x} & \mathbf{M}_{dyn,u} \cdots \\ \mathbf{M}_{cons,x} & \mathbf{M}_{cons,x} & \mathbf{M}_{cons,u} \cdots \\ \vdots & \vdots & \vdots \\ \mathbf{M}_{cons,x} & \mathbf{M}_{cons,x} & \mathbf{M}_{cons,u} \cdots \end{array}\right] \cdots$$

$$\left[\begin{array}{ccccc} \cdots & OR(\mathbf{M}_{cost_{\Phi,x}}, \mathbf{M}_{cost_{\Psi,x}}) & OR(\mathbf{M}_{cost_{\Phi,x}}, \mathbf{M}_{cost_{\Psi,x}}) & & \\ \cdots & \mathbf{M}_{dyn,x} & \mathbf{M}_{dyn,u} & & \\ \vdots & \vdots & \vdots & & \\ \cdots & \mathbf{M}_{dyn,x} & \mathbf{M}_{dyn,u} & \mathbf{0}_{[N\cdot(n_s+n_g)+1\times 1]} & \\ \cdots & \mathbf{M}_{cons,x} & \mathbf{M}_{cons,u} & & \\ \vdots & \vdots & \vdots & \vdots & \\ \cdots & \mathbf{M}_{cons,x} & \mathbf{M}_{cons,u} & & \end{array}\right] \tag{7.90}$$

and the operator $\circ$ represents the Hadamard product, that is the product element by element. Given two matrices $\mathbf{A}$ and $\mathbf{B}$, having dimensions $m \times n$, their Hadamard product $\mathbf{C} = \mathbf{A} \circ \mathbf{B}$ is

$$\mathbf{C}_{i,j} = \mathbf{A}_{i,j}\mathbf{B}_{i,j},\ i = 1, \cdots, m;\ j = 1, \cdots, n; \tag{7.91}$$

It will then be possible to apply the knowledge acquired from the continuous functions to know which elements must be computed numerically (i.e., the non-zero elements of the matrix $\mathbf{M}_{GD}$).

7.4.2.4 Theoretical Jacobian

Finally, a third contribution, the theoretical Jacobian, arises in case we deal with problems having an unknown final time. The NLP state vector will then have a further variable, which is t_F. In this case, the Jacobian associated to this term is proportional to the output of the continuous functions in virtue of the time mapping reported in Eqs. (7.19) and (7.20).

$$\mathbf{Jac}_{Th} = -\frac{1}{2}\begin{bmatrix} & & 0 \\ & & \mathbf{f}_1 \\ & & \mathbf{f}_2 \\ \mathbf{0}_{[N\cdot(n_s+n_g)+1]\times[(N+1)\cdot n_s+N\cdot n_c+1]} & & .. \\ & & \mathbf{f}_N \\ & & \mathbf{0}_{N\cdot n_g\times 1} \end{bmatrix} \tag{7.92}$$

The hybridization of the Jacobian matrix has as consequence that only a small part (4.68% and 8.84%, for 25 and 100 nodes, respectively) of its elements need numerical computation techniques. Hence, significant CPU time is saved when solving the optimal control problem, or more specifically, the NLP problem representing it. To see which are the results, let us see two significant examples, already used as examples for testing the automatic scaling routines. The solutions have been generated by using 25, 50, 75, and 100 nodes.

7.4.3 CPU Time Comparison

Here, a comparison between the CPU time required to compute the solution of the full nonlinear Jacobian NLP problems and the hybrid Jacobian problems has been performed. In particular, in Tables 7.8 and 7.9 four cases for each of the problems considered are compared. All the simulations have been performed with a laptop having an i7M640 CPU with a clock frequency of 2.80 GHz and 4 GB of RAM. All the examples considered do not have any meaningful initial guess from previous solutions. In other words, all the problems were solved with cold start.

First and second columns report the time required for generating a solution, while the third column reports the time ratio expressed as percentage between the CPU time required to solve the problem with hybridization, and the time required to do it without hybridization. The fourth column shows the offline time, which is common to the two methods. It can be observed how the time is strongly reduced (in some cases it becomes the 4% of the initial CPU time for the orbit raising, and 5.8% for the Space Shuttle problem.

It is clear that the hybridization of the Jacobian matrix yields an increase in the speed of the computation of the NLP solutions. In the cases analysed, the improvement in the CPU time is between 11 and 17 times. Additionally, their quality is improved as well, as a consequence of the higher accuracy of the computation of the Jacobian matrix defining the NLP problem. Indeed, the non-exact computation of the Jacobian is reduced

Table 7.8: CPU time required for orbit raising problem.

Nodes	CPU Time [s] (Full)	CPU Time [s] (Hybrid)	CPU Time Ratio (%)	Offline Time [s]
25	11.15	0.45	4.05	0.37
50	22.08	1.55	7.04	0.71
75	71.49	2.63	3.68	1.76
100	202.32	12.29	6.07	5.73

Table 7.9: CPU time required for the Space Shuttle reentry problem.

Nodes	CPU Time [s] (Full)	CPU Time [s] (Hybrid)	CPU Time Ratio (%)	Offline Time [s]
25	15.11	0.88	5.83	0.61
50	42.84	4.54	10.60	1.43
75	171.65	18.29	10.65	3.92
100	594.17	50.56	8.51	10.83

to the numerical contribution, and the use of the complex-step for its computation represents a very good approximation.

7.5 Conversion of the solution of the NLP problem into a continuous form

At the end of the computation of the NLP solution, it is straightforward to convert the obtained discrete information into a continuous form. This can be done using Eqs. (7.11) and (7.12). While for the states the initial conditions are usually an input, for the controls we can distinguish two possibilities.

- The initial control is known;
- The initial control is unknown;

In the former case, the Eq. (7.12) is replaced by (7.93).

$$\mathbf{u}(t) \cong \sum_{i=0}^{N} \mathbf{U_i} P_i^u(t) \tag{7.93}$$

The computation of the continuous controls becomes mathematically equivalent to the computation of the continuous states performed through Eq. (7.11). In the latter case, Eq. (7.12) is used, and the initial controls are simply extrapolated by evaluating Eq. (7.12) at time t_0.

7.6 Validation through literature examples

Now that the entire transcription process of SPARTAN has been described, the validation of the tool can be performed via comparison with literature results. In this subsection this comparison is reported. Specifically, two examples are included. The former is the orbit raising problem, treated by Herman and Conway[67], while the latter is the Space Shuttle reentry problem, which has already been defined in this chapter.

7.6.1 Orbit Raising Problem

For the orbit raising there is no contribution coming from the theoretical Jacobian matrix, since the final time is known. The Jacobian here will only consist of the pseudospectral (starred elements) and the numerical contributions (circles for the differential equations, dark star for the cost function). All the figures showing states, controls and constraints are associated with the solutions obtained using 100 nodes, while the Jacobian structure is derived from a NLP having 5 nodes (chosen for a better visualization of the patterns).

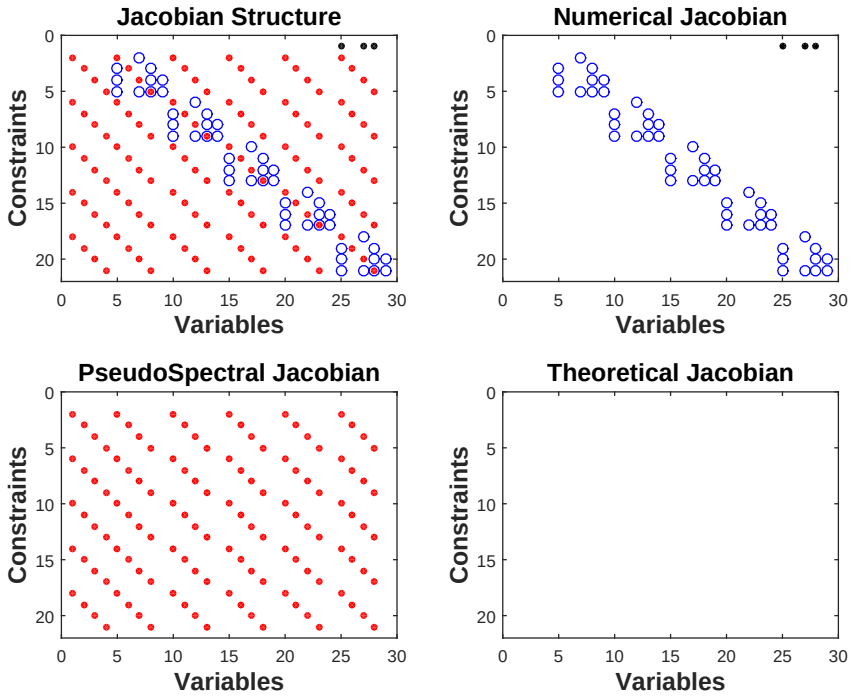


Figure 7.9: Jacobian matrix sparsity patterns for orbit raising problem.

The pseudospectral solution to this problem is represented in Figs. 7.10, and 7.12. These results can be compared with literature results obtained by Herman and Conway[67], shown in Figs. 7.11, 7.13.

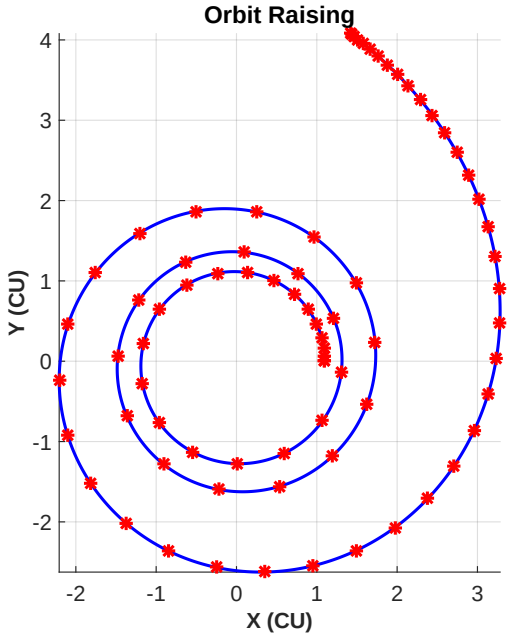


Figure 7.10: In-plane orbit evolution.

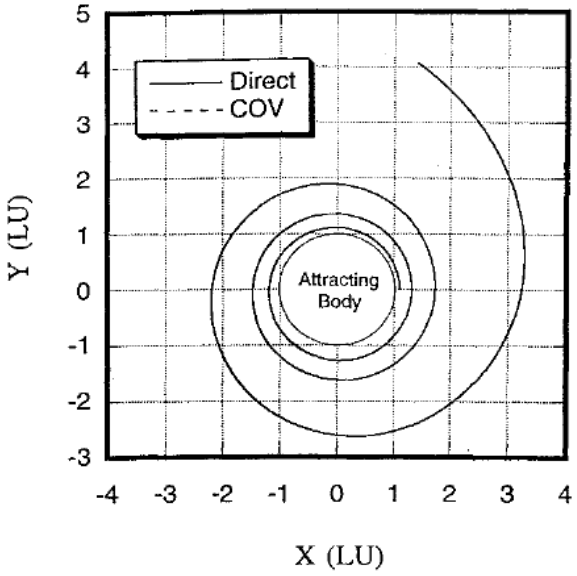


Figure 7.11: In-plane orbit evolution[67].

.

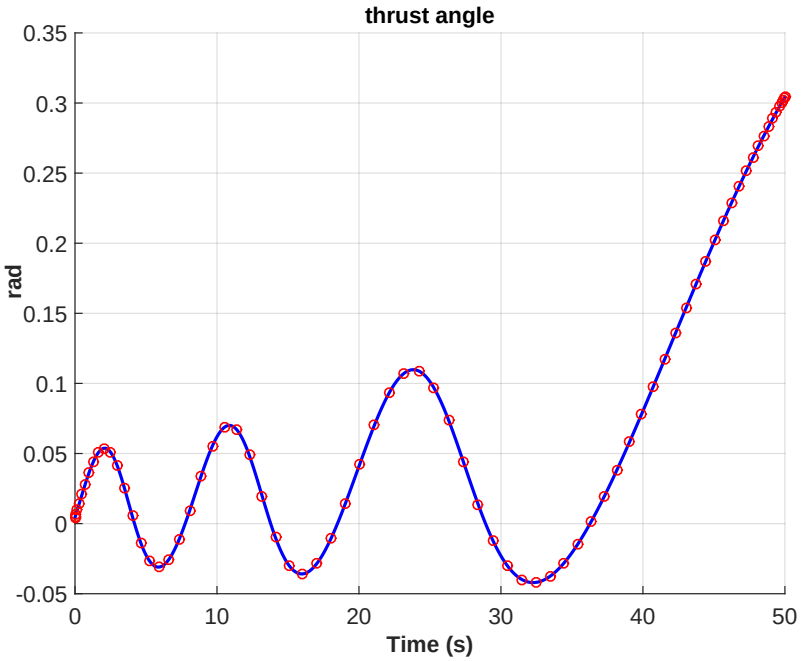


Figure 7.12: Control evolution for orbit raising problem.

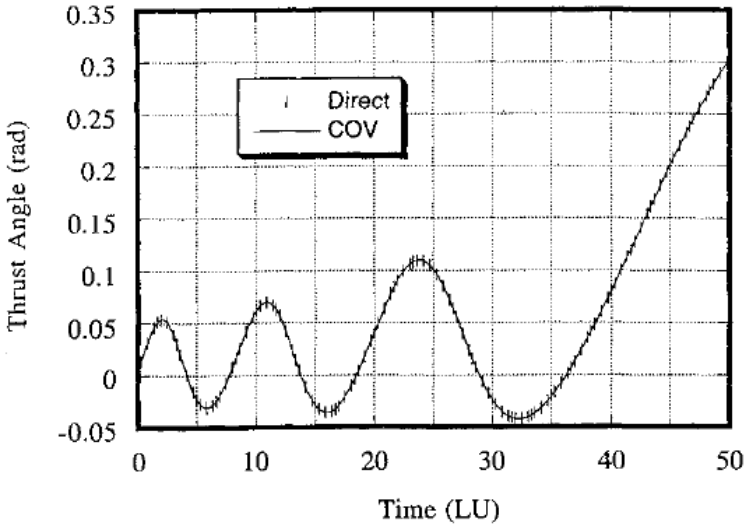


Figure 7.13: Control evolution for orbit raising problem[67].

.

7.6.2 Space Shuttle Reentry Problem

In this case the final time is unknown, and the Jacobian has all the three components (small-starred for pseudospectral, circle, light-starred, and dark-starred for numerical contributions of dynamics, constraints and cost function, respectively, and light-starred, vertical oriented for the theoretical contributions) presented in this section. The patterns representing the Jacobian are shown in Fig. 7.14. Figures 7.15, 7.17, and 7.19 show the states, the controls and the heat flux, respectively. Results can be compared with the ones obtained by Betts, and plotted (in dotted lines) in Figs. 7.16, 7.18, and 7.20.

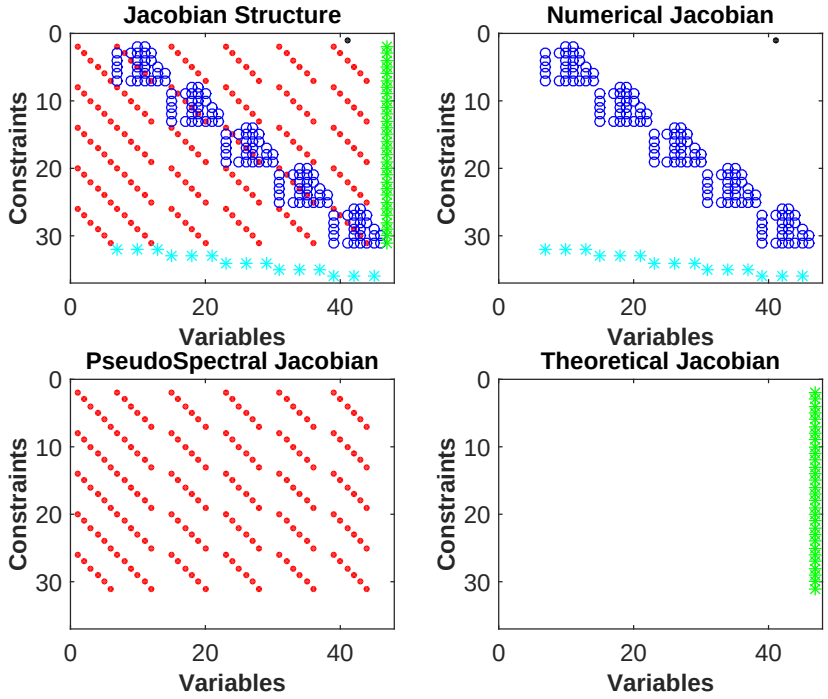


Figure 7.14: Jacobian matrix sparsity patterns for the Space Shuttle reentry problem.

It is possible to see that the results are fully consistent. The values of the cost functions for both the problems are also equal to the ones reported in literature. It is now possible to see how SPARTAN has been used to generate optimal trajectories for SHEFEX-3.

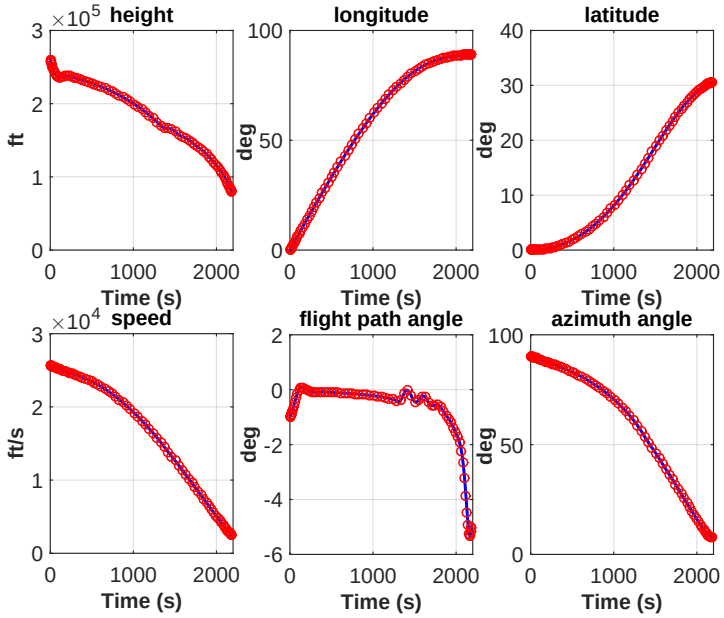


Figure 7.15: States evolution the Space Shuttle reentry problem.

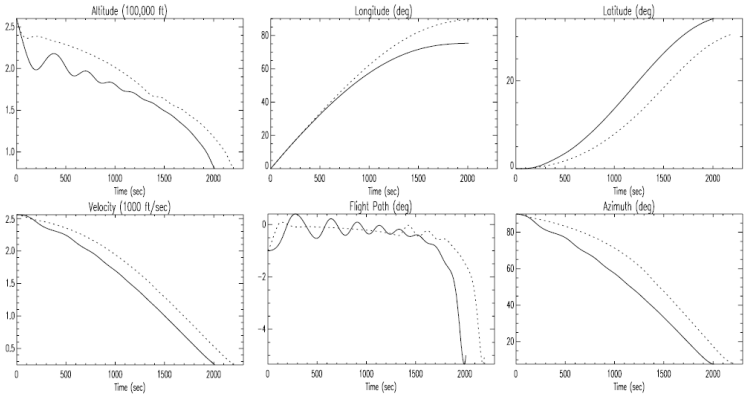


Figure 7.16: States evolution the Space Shuttle reentry problem[58] - dotted lines.

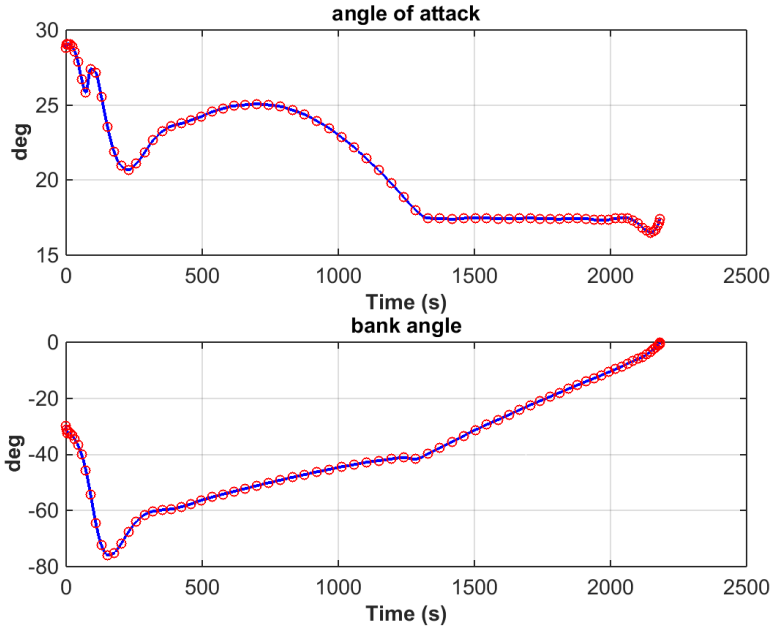


Figure 7.17: Controls evolution the Space Shuttle reentry problem.

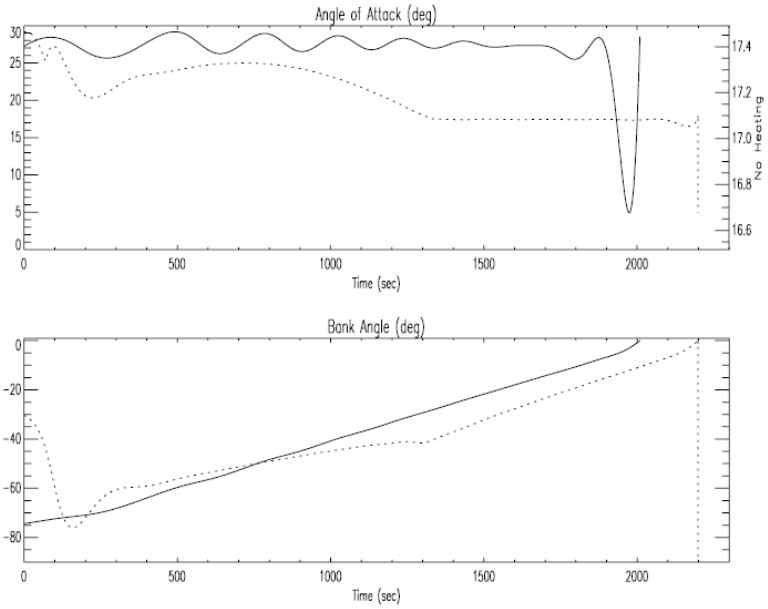


Figure 7.18: Controls evolution the Space Shuttle reentry problem[58] - dotted lines.

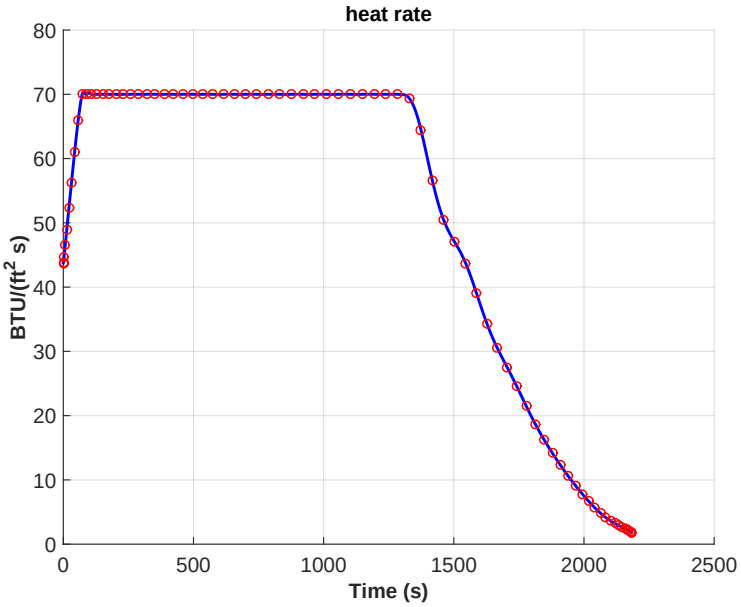


Figure 7.19: Heat flux evolution for space shuttle reentry problem.

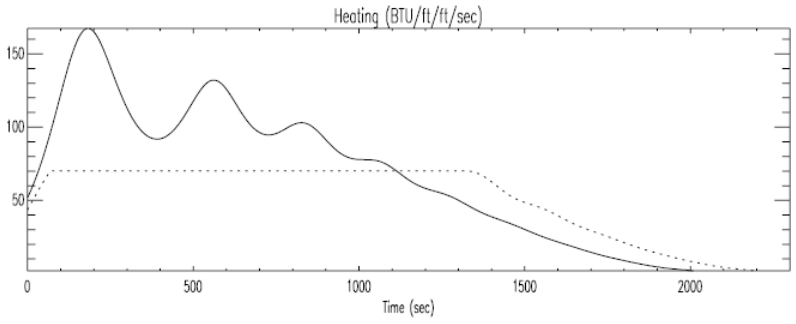


Figure 7.20: Heat Flux evolution for space shuttle reentry problem[58] - dotted line.

Chapter 8

Adaptive Feedforward Guidance

In the previous chapter the transcription process to generate optimal trajectories has been explained. An optimal control problem can be setup and solved, having as a result the reference states, and the reference controls corresponding to a given set of initial states. However, during the real mission, several variables will differ from the nominal ones. Variables, like atmospheric density and temperature, can still show differences w.r.t. the models used to represent them. Moreover, aerodynamic coefficients and mass can be slightly different w.r.t. the nominal values. Other uncertainties come from dispersions at the entry interface, as a consequence of variation in the launch conditions and ascent trajectory. Finally, wind can affect the flight.

To take these effects into account, two strategies have been developed. We have to make a distinction between known and unknown uncertainties, where the adjectives "known" and "unknown" are meant at the moment of the real flight. In the first category we can include the dispersion at the entry interface, that is, the three components of position and the three components of velocity. The navigation system will provide them with good accuracy, therefore they can be considered known uncertainties. Atmospheric variations, and aerodynamics uncertainties, mass at the entry interface and wind disturbances are all bounded, but unknown factors playing a role during the entry phase. To deal with all these uncertainties, a feedforward-feedback guidance scheme is proposed. The feedforward guidance scheme, explained in this chapter, will compensate for known errors (i.e., initial errors in position and velocity), while the development of the feedback control law will be the subject of the Chapter 9. The feedforward guidance solution is generated by combining online the solutions belonging to a database of optimal trajectories, which were previously computed offline with SPARTAN, and stored onboard. The proposed method for combining the trajectories takes the name of adaptive multivariate pseudospectral interpolation (AMPI).

A practical example of the method for a 1-D case is reported in Sec. 8.1, while a formal mathematical definition of the method, together with the algorithms to realize it, is described from Sec. 8.2 through Sec. 8.6. Finally, the results associated with a reduced number of simulation are reported in Sec. 8.7, to show and quantify the advantages of using the proposed scheme. The adaptive feedforward guidance presented

here provides the reference trajectory and the controls to track it even in presence of significant variations of the states, and can be seen as an onboard trajectory-generator algorithm. The working scheme of the adaptive feedforward guidance algorithm is depicted in Fig. 8.1.

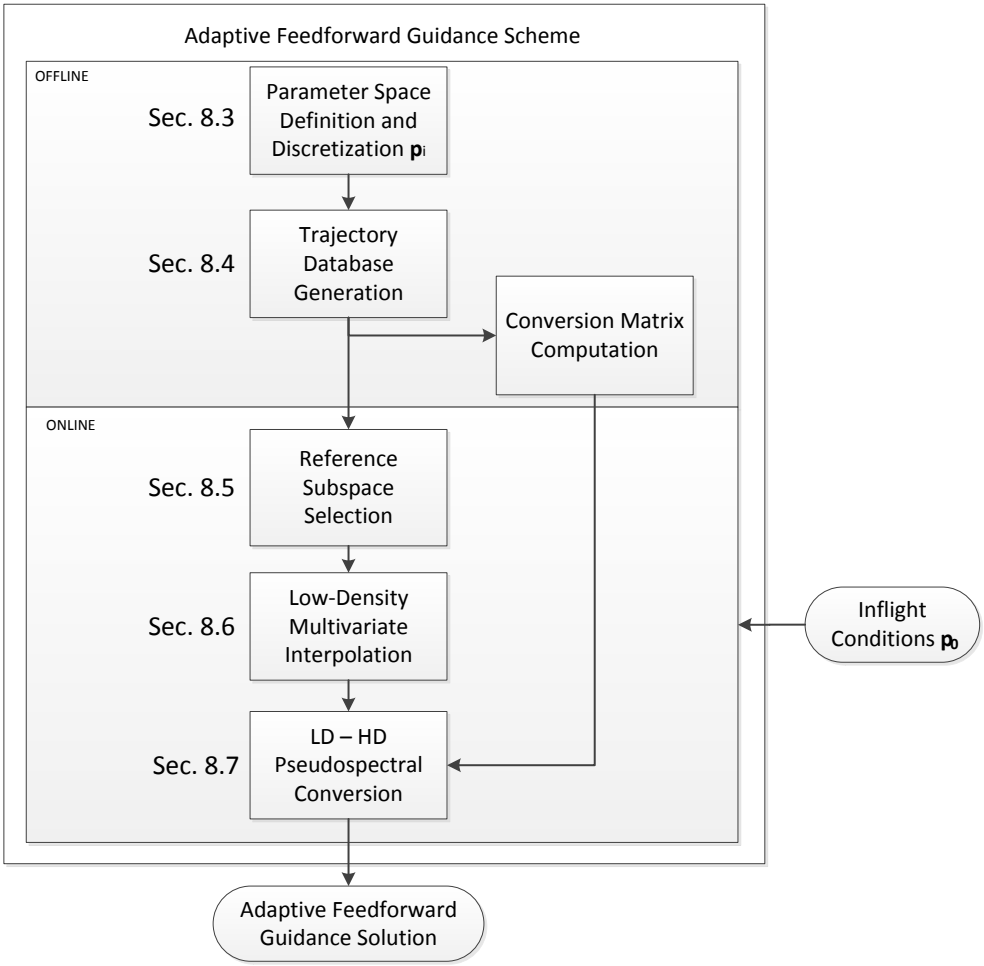


Figure 8.1: Scheme of Adaptive Multivariate Pseudospectral Interpolation.

We can look at the scheme shown in Fig. 8.1 as a part computed offline, and a part computed online. The offline part involves the identification of the parameters, which can be off-nominal, and determined during the flight (e.g., the states at the entry interface, provided by the navigation subsystem); under these hypotheses it is possible to apply the second part of the AMPI, which will run online. A specific range for each of the parameters needs to be determined and sampled, resulting in a series of discrete parameters $\mathbf{p_i}$. Accordingly to these parameters, a corresponding series of parametric

optimal-control problems is solved. This will result in the trajectory database to be stored online.

A further output of the trajectory-database generation is the LD-HD conversion matrix, used, as the name suggests, to convert the low-density (LD) trajectory (less stringent in terms of on-board memory requirements) into a more meaningful high-density (HD) solution, with a process of loss-less conversion, as we will see. During the mission, the inflight parameters, $\mathbf{p_0}$, different w.r.t. the nominal ones, will be analysed to select the reference subspace over the entire trajectory database.

The selected subspace will provide the basis to perform a multivariate interpolation process to compute the adaptive LD solution. Finally, the previously computed LD-HD conversion matrix is used to convert the HD solution, which represents our adaptive feedforward guidance solution. All these aspects will be explained in detail in the next sections of this chapter. However, to make the process clearer, a one-dimensional example will be given in the next section. This model will only provide a practical example, while in the rest of the chapter, the mathematical details for the extension of the method to d dimensions will be given.

8.1 Adaptive Multivariate Pseudospectral Interpolation - 1D Example

Let us consider the following example. We are interested to minimize the cost function

$$J = |x(t_f)| \tag{8.1}$$

of a one-dimensional system subject to the following nonlinear differential equation

$$\dot{x}(t) = -x + xu - u^2 \tag{8.2}$$

where x being the state, and u being an (as yet) unspecified control. Let us apply the procedure defined in Fig. 8.1 to this problem, similar to what has been proposed by Garg[59]. The first step is the discretization of the parameter space. These points are represented in Fig. 8.2. In this simplified example, the only parameter is the initial condition $x(t_0)$, sampled in the points -0.5, 0.5, 1.5, and 2.5. With the application of the AMPI scheme, we will be able to generate a trajectory having any initial condition $x_0 \in [-0.5, 2.5]$. The following step is the trajectory database. We will solve three optimal control problems, which differ from each other only in terms of the initial condition. In Fig. 8.2 we can see the continuous black lines representing the optimal-control trajectories corresponding to the selected initial conditions. The black starred points are the initial states used to compute the trajectory database of optimal solutions. Suppose we want to compute a solution corresponding to the initial condition $x_0 = 1$ (The red circle in Fig. 8.2).

With the reference subspace algorithm we will determine which of the three possible intervals ($x_0 \in [-0.5, 2.5]$, $x_0 \in [0.5, 1.5]$, and $x_0 \in [1.5, 2.5]$) is associated with the current value of our parameter x_0. The application of the algorithm 1 will give us the

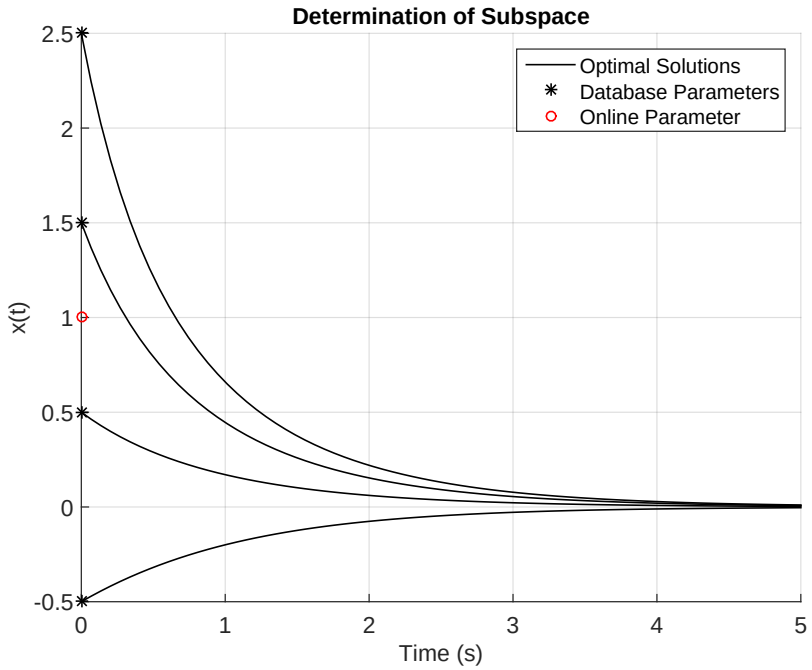


Figure 8.2: Example of interpolation scheme for 1-D nonlinear problem - optimal trajectories corresponding to different initial conditions.

interval $[0.5, 1.5]$, (the gray area in Figs. 8.3, 8.4), as it is shown in Table 8.1. In this case x_i is equal to 1, while the vector $\mathbf{p}^i$ is equal to $[2.5,\ \ 1.5,\ \ 0.5,\ \ -0.5]$. In table 8.1 the results element by element are reported. From Table 8.1 it is possible to see how

Table 8.1: Numerical application of algorithm 1.

idx	p_i	$p^i - x_i$	$\mathbf{eP}_i$	$\mathbf{\Delta p}^i_j$
1	2.5	1.5	1	0
2	1.5	0.5	1	2
3	0.5	-0.5	-1	0
4	-0.5	-1.5	-1	-

the algorithm works. The analysis of the intervals leads to have the indeces 2 and 3 (enlighted in the table) as the ones enclosing the off-nominal parameters.

Let us look at the interpolation process as a black-box, which needs to be evaluated a certain number of times. With reference to Fig. 8.5, suppose we want to have a meaningful solution (the so-called high-density discrete solution) in $N_{HD}+1$ nodes. The names "high-density" (and "low density", introduced later) only refer to the number

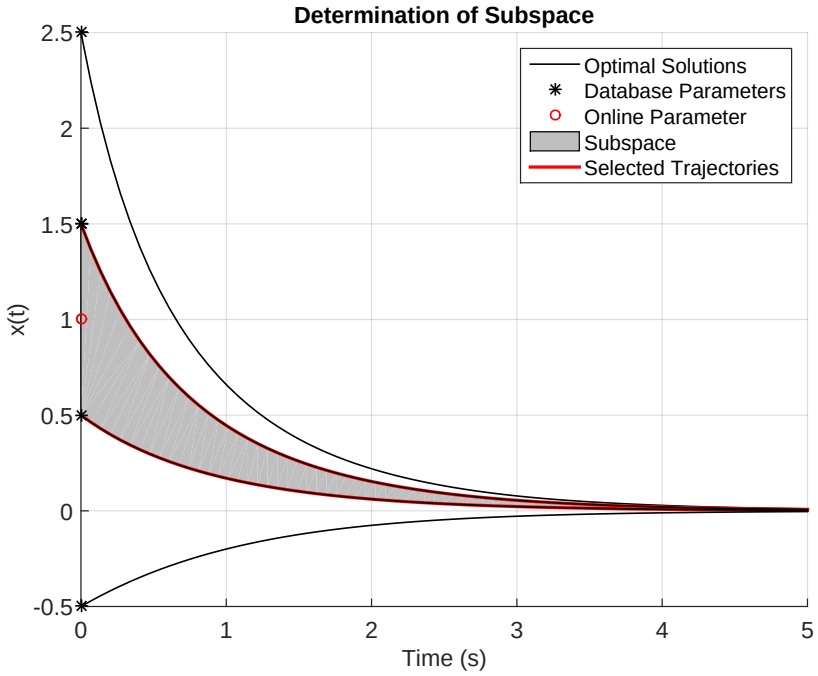


Figure 8.3: Example of interpolation scheme for 1-D nonlinear problem - selected subspace.

of nodes used to compute the trajectory over a given domain, and has nothing to do with the atmospheric density. The +1 refers to the inclusion of the initial node. In the case considered here the solutions have $50 + 1$ nodes. An approximated solution can be found by applying the interpolation algorithm at each of the nodes; this results in the $50 + 1$ nodes (represented as red circles), which form our interpolated solution.

An alternative scheme is depicted in Fig. 8.6. In this case we are exploiting the aforementioned properties of the pseudospectral methods to evaluate the interpolation operation at a significantly smaller number of nodes, which will be referred as the $N_{LD} + 1$ nodes, that is, the so-called low-density discrete solution. In the proposed case, only $10 + 1$ nodes were used. Since in complex problems like atmospheric entry, large numbers of HD nodes need to be used to capture the behavior of the system, we need to convert the LD trajectory into its corresponding HD trajectory. An example is depicted in Fig. 8.7, where the LD representation of the state $x(t)$ (the red circles) is converted into a HD representation (the green squares). This solution is the on-board trajectory generated online, which represents an adapted solution to the off-nominal conditions experienced during the flight.

To give an idea about the results, in Fig. 8.8 the states obtained by solving the optimal control problem associated with $x_0 = 1$, and by applying the AMPI scheme, are plotted. In this specific case the error between the true optimal solution and the interpolated solution is less than 4%. Moreover, with this technique the entire space defined by the trajectory database can be covered (the dark gray area in Fig. 8.8). This

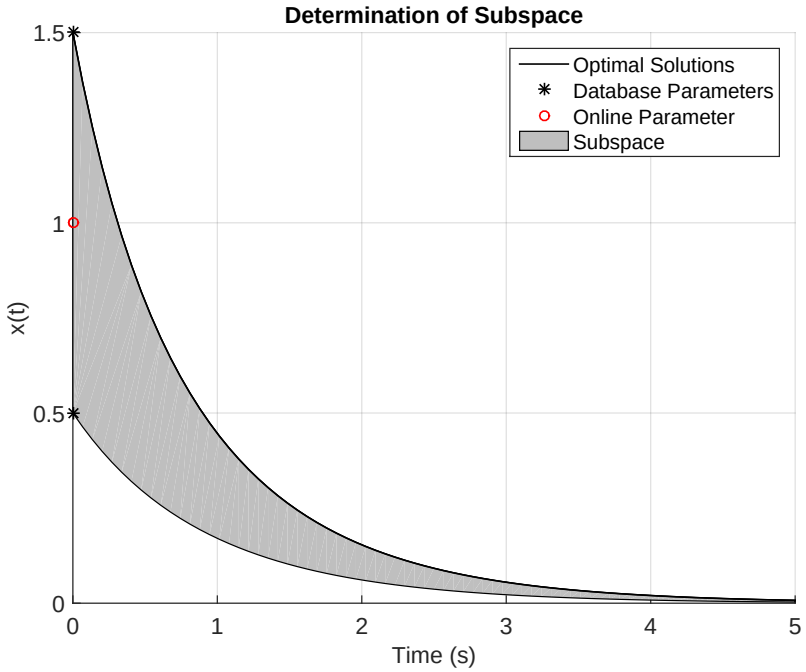


Figure 8.4: Example of interpolation scheme for 1-D nonlinear problem - selected subspace.

approach leads to a significant decrease of the CPU burden required to compute an interpolated solution, as the total number of interpolations is strongly reduced. Another advantage is that the size of the database is strongly reduced as well (in the case of SHEFEX-3 the reduction can be up to 96% of the original size. Figures 8.9 and 8.10 show the differences in the two databases. While in Fig. 8.9 a large number of points for each trajectory needs to be stored, in Fig. 8.10 only the points in the LD nodes need to be kept in memory.

Moreover, the proposed algorithm is lossless, which means that the results depicted in Figs. 8.5 and 8.6 are identical and there is no loss of information, even if the number of the interpolations in the latter case is strongly reduced. This is a consequence of using nodes obtained by Radau polynomials for sampling the trajectory-database. This is a strong implication of the proposed method, as it can be coupled to any method used to compute optimal solutions, and not necessarily to pseudospectral methods. Another strong implication is that the method is real-time capable, as it is possible to predict the exact number of operations required to generate on-board a complete trajectory.

In general one can see that, since we are dealing with nonlinear systems, the error will increase when the parameter discretization becomes coarser, but with the proper selection of the grid, satisfying results are obtained. The selection of the parameter space is therefore a heuristic process based on a trade-off between the size of the database, which increases when the number of discrete points, where to compute the optimal trajectories, increases, and the accuracy of the interpolated scheme, which

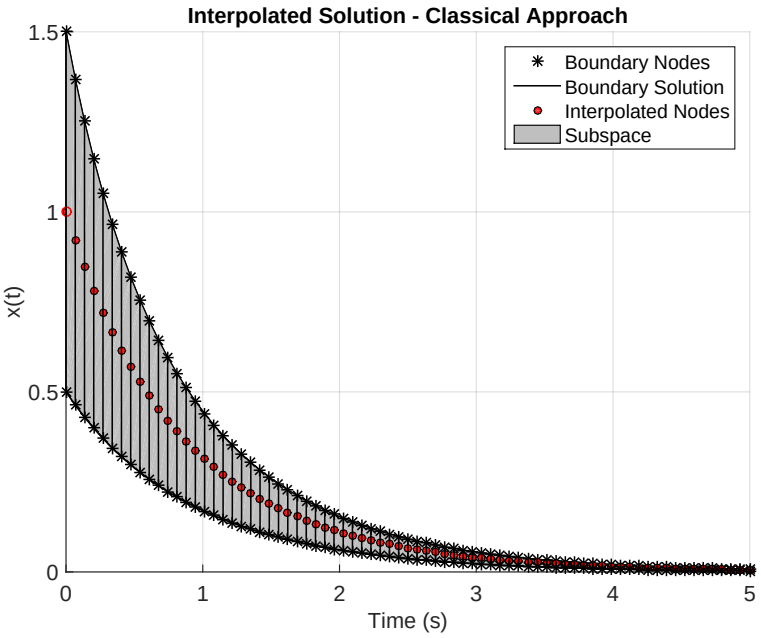


Figure 8.5: Example of interpolation scheme for 1-D nonlinear problem - classical interpolation scheme.

decreases when the number of discrete points decreases. We can now extend the method introduced here and extend it to d dimensions, providing formal definitions at the same time.

8.2 Definition and discretization of the parameter space

The first step is the proper definition of the parameter space. In this context we will consider as parameters the entry interface conditions, provided by the navigation solution. This is not the only choice, since every parameter that can be estimated on-board and that allows to compute a corresponding optimal solution can be potentially treated with this method. In this case, we consider six different off-nominal initial conditions (three components for the position, and three components for the velocity), but to keep the method general, let us refer to d parameters as $p^1, \dots, p^d$. Each of the parameters is defined on a compact subset of the real numbers $\mathbb{R}_i \subset \mathbb{R}, i = 1, \dots, d$, such that the Cartesian product

$$\mathbb{R}^d \supset \mathcal{P}_c = \prod_{i=1}^{d} \mathbb{R}_i = \mathbb{R}_1 \times \cdots \times \mathbb{R}_d, \tag{8.3}$$

defines the parameter space $\mathcal{P}_c$. In this context a solution of the OCP can be interpreted not only as a function of time, but also as function of a d-dimensional parameter

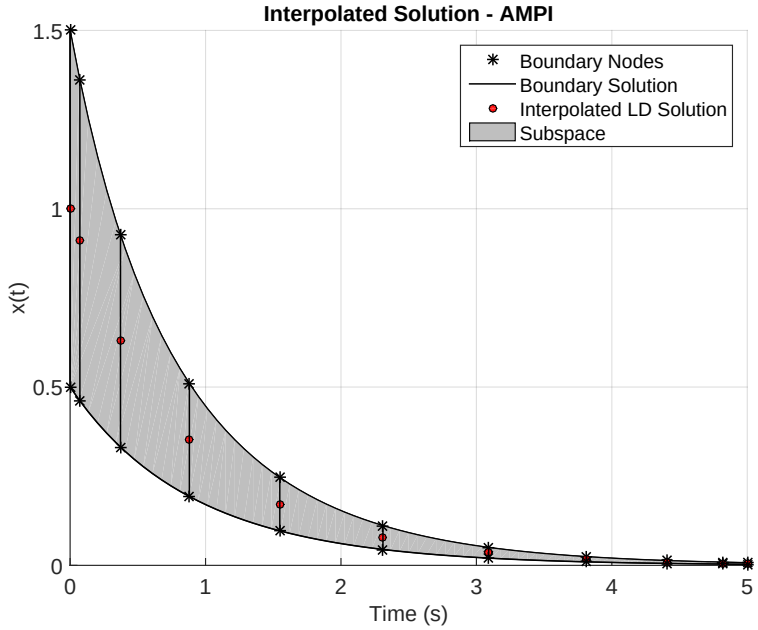


Figure 8.6: Example of interpolation scheme for 1-D nonlinear problem - pseudospectral interpolation scheme.

vector. In the following each one-dimensional domain of a single reference parameter is discretized using a finite number of discrete points. Therefore, let

$$\mathbf{p}^i = \left\{p_1^i, \ldots, p_{n_i}^i\right\} \tag{8.4}$$

define a strictly monotonically increasing set for each $i = 1, \ldots, d$. The Cartesian product of the sets given by Eq. (8.4) defines a d-dimensional $n_1 \times \cdots \times n_d$-rectangular grid

$$\mathcal{P} = \prod_{i=1}^{d} \mathbf{p}^i = \mathbf{p}^1 \times \cdots \times \mathbf{p}^d \tag{8.5}$$

which can be seen as a discretization of the parameter space $\mathcal{P}_c$ defined by Eq. (8.3). The set $\mathcal{P}$ consists of $n_G = \prod_{i=1}^{d} n_i$ elements and can equivalently be represented as a combination of all the grid points $_i$, where $_i = \left(p_{i_1}^1, \ldots, p_{i_d}^d\right) \in \mathbb{R}^d$, such that

$$\mathcal{P}_c \cong \mathcal{P} = \{\mathbf{p_i}\}_{i_1=1,\ldots,i_d=1}^{n_1,\ldots,n_d} . \tag{8.6}$$

The parameter space is therefore completely defined. For each parameter, the related OCP can be redefined and solved. The result will be a set of parameters, which cover the entire d-dimensional space $\mathcal{P}_c$. A practical example of d-dimensional parameter space, with $d = 6$, is shown in Eq. (8.7).

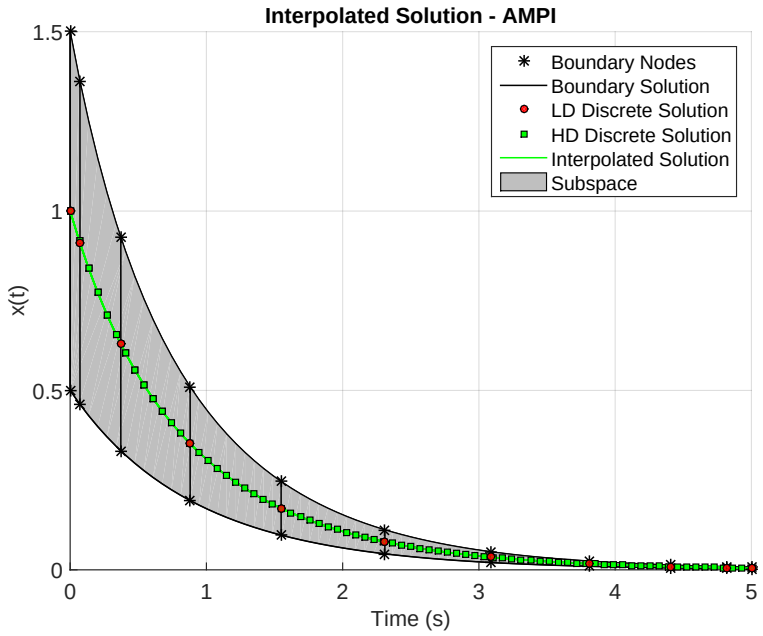


Figure 8.7: Example of interpolation scheme for 1-D nonlinear problem - conversion from LD to HD representation.

8.3 Generation of Trajectory-Database

Once the mathematical tools needed for the online trajectory generation have been described, we can apply them to a trajectory database. Therefore, the following problem is the trajectory-database generation. For a complex mission, such as the atmospheric entry, the inflight conditions can be significantly different w.r.t. the nominal ones, and this aspect directly affects the database size. Therefore, the reference parameters, which could vary need to be identified. The driving idea is to perform a mapping of the range we want to cover, and to generate a database of trajectories, which fulfill the requirements defined in Chapter 2 for the entire region of interest. This information will then be processed online and used to adapt the feedforward guidance to the current inflight conditions.

Let us now define the parameter space for our case. We can characterize the uncertainty on the initial states from a purely geometrical point of view. Indeed, a 1-D region of interest X can be represented as a straight line connecting two nodes representing the extreme values that this particular variable can assume, see Fig. 8.11(a). This is the case treated in the example of Section 8.1. The extension of this region to two dimensions X, Y is geometrically represented by a rectangle (or in an easier way, by a square if the variables are properly normalized), where the vertices are the 2^2 possible combinations of extreme values that the variables X and Y can assume, as depicted in Fig. 8.11(b). In three dimensions X, Y, Z, we will have a cube, which vertices represent

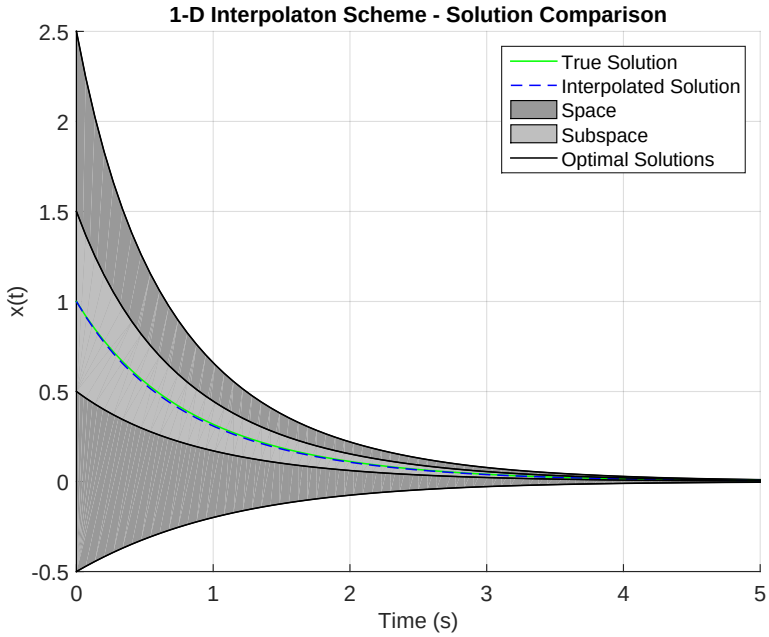


Figure 8.8: Example of interpolation scheme for 1-D nonlinear problem - true vs interpolated solution.

the 2^3 possible combinations of parameters (Fig. 8.11(c)). Since the initial state of the vehicle at the entry interface is represented by the three components of position and the three components of speed, we will have a six-dimensional region of interest, which can be different from their corresponding nominal values. We can describe this multidimensional uncertainty as a series of *hexeracts*, (depicted in Fig. 8.11(d)), which is a member of the hypercubes family, characterized by having a dimension equal to six.

The hypervertices of the selected hexeract represent the extreme initial conditions that the vehicle can experience at the entry interface, while its hypervolume encloses all the possible intermediate initial states. The nominal conditions for SHEFEX-3 will represent then the hypercenter of our hexeract. According to the information provided by the responsible team for the Launch and Ascent trajectory, the following parameter space for the entry interface have been defined.

$$\begin{aligned}
\mathbf{p}_1 &= \delta h = [-250, 0, +250] \text{ m} \\
\mathbf{p}_2 &= \delta\theta = [-0.5, 0, +0.5] \text{ deg} \\
\mathbf{p}_3 &= \delta\phi = [-0.5, 0, +0.5] \text{ deg} \\
\mathbf{p}_4 &= \delta V = [-70, 0, +70] \text{ m/s} \\
\mathbf{p}_5 &= \delta\gamma = [-0.5, 0, +0.5] \text{ deg} \\
\mathbf{p}_6 &= \delta\psi = [-0.5, 0, +0.5] \text{ deg}
\end{aligned} \tag{8.7}$$

With these definitions, it is now possible to set and solve the parametric optimal-control

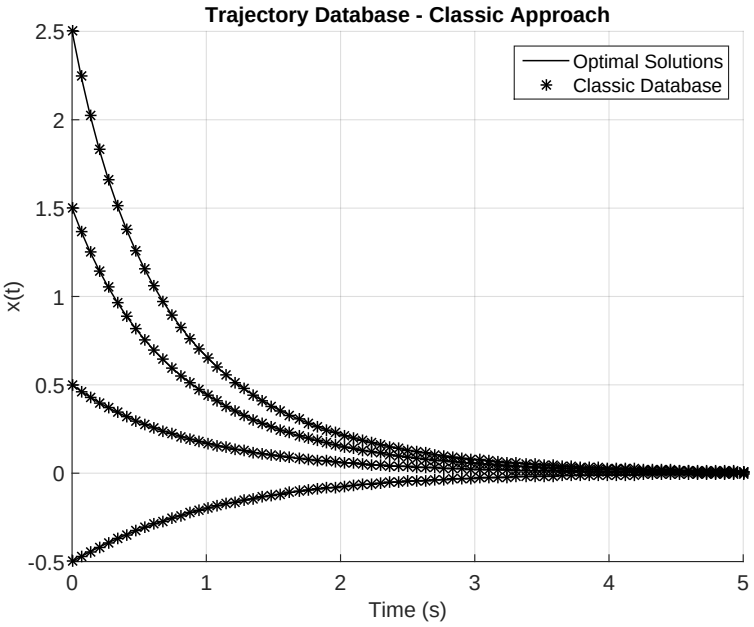


Figure 8.9: Trajectory Database - Classic approach.

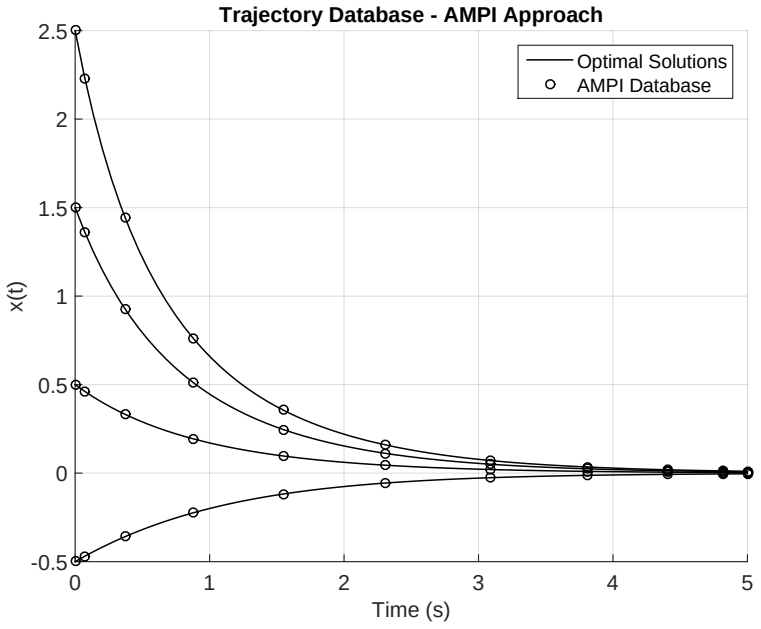
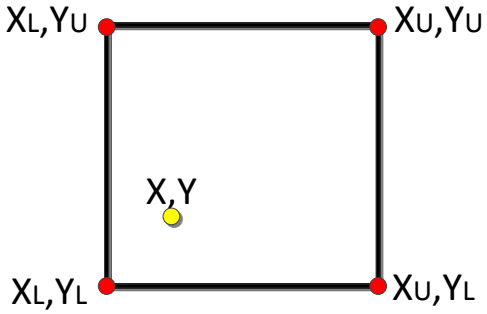
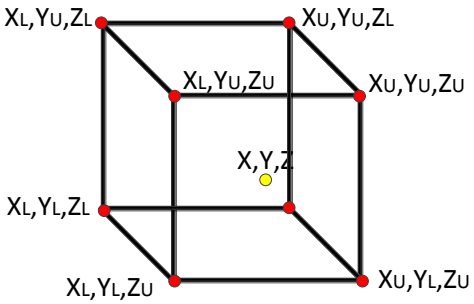


Figure 8.10: Trajectory Database - AMPI approach.

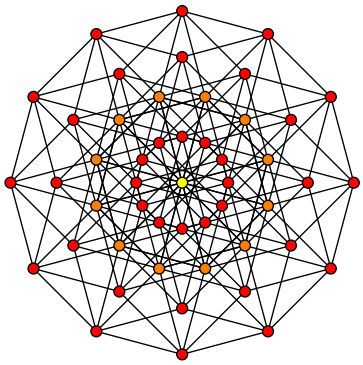
(a) Example of 1-D Uncertainty.



(b) Example of 2-D Uncertainty.



(c) Example of 3-D Uncertainty.



(d) Example of 6-D Uncertainty.

Figure 8.11: Examples of multidimensional uncertainties.

problem, which can be stated as follows.

The objective of the optimal-control problem is to minimize the cost function J

$$J = w_\theta(\theta_F - \theta_{ref})^2 + w_\phi(\phi_F - \phi_{ref})^2 + w_h(h_F - h_{ref})^2 + w_V(V_F - V_{ref})^2 \qquad (8.8)$$

subject to the differential equations defined in Eq. (4.5). The states are bounded:

$$\begin{bmatrix} 0\text{ km} \\ -180^{\text{o}} \\ -90^{\text{o}} \\ 10\text{ m/s} \\ -45^{\text{o}} \\ -180^{\text{o}} \end{bmatrix} \leq \begin{bmatrix} h \\ \theta \\ \phi \\ V \\ \gamma \\ \psi \end{bmatrix} \leq \begin{bmatrix} 120\text{ km} \\ 180^{\text{o}} \\ 90^{\text{o}} \\ 7000\text{ m/s} \\ 30^{\text{o}} \\ 180^{\text{o}} \end{bmatrix} \tag{8.9}$$

and the controls satisfy the requirements reported in Table 2.2. Three constraints are included in the transcription, that is, the dynamic pressure, $\overline{q}$, the heat flux, $\dot{Q}$ (computed by using the cold-wall model for a laminar boundary layer), and the vertical load factor, n_z. These constraints are computed according to

$$\begin{aligned} \overline{q} &= \frac{1}{2}\rho V^2 \\ \dot{Q} &= k_q\sqrt{\rho}V^3 \\ n_z &= \frac{|L\cos\alpha + D\sin\alpha|}{g_0} \end{aligned} \tag{8.10}$$

where ρ is the atmospheric density, expressed in kg/m^3, k_q is a constant depending on the material and the geometry of the thermal protection system, for SHEFEX-3 equal to $1.2444 \cdot 10^{-3}$ kg$^{1/2}$/m^3, and g_0 is the gravitational acceleration at sea level, $g_0 = 9.782$ m/s^2. For these constraints, limits are reported below, and are consistent with the requirements defined in Table 2.4.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} \overline{q} \\ \dot{Q} \\ n_z \end{bmatrix} \leq \begin{bmatrix} 5 \cdot 10^4\text{ N/m}^2 \\ 6.5\text{ MW/m}^2 \\ 10 \end{bmatrix} \tag{8.11}$$

The initial conditions are parametrized according to the conditions reported in Eq. (8.7),

$$\mathbf{x}(t_0) = \mathbf{x}^*(t_0) + \begin{bmatrix} \delta h_i \\ \delta\theta_j \\ \delta\phi_k \\ \delta V_l \\ \delta\gamma_m \\ \delta\psi_n \end{bmatrix}, \qquad i,j,k,l,m,n = [1,2,3] \tag{8.12}$$

where $\mathbf{x}^*(t_0)$ is the nominal entry interface, reported in Table 2.3. One can see from Eq. (8.8) that the terminal condition $M = 2$ has been indirectly included through the terminal values of altitude and velocity as a soft constraint in the cost function, i.e., a reference final altitude of 23.5 km, and a reference final velocity of 595 m/s. It is therefore possible to limit not only the Mach number, but also the dispersions on altitude and velocity according to the requirements. The weights are assumed to be equal to 1 for the longitude and the latitude, and equal to 10^{-4} for the altitude and

the velocity. In total, 3^6 trajectories have been computed, for a total of 2^6 hexeracts. Each of the computed 3^6 trajectories has been formulated according to the OCP defined in Eqs. (8.8)-(8.12). The states and the controls evolution for the entire database are depicted in Figs. 8.12 and 8.13, while the envelope of the trajectories is shown in Fig. 8.14. Finally, the constraints are illustrated in Fig. 8.15.

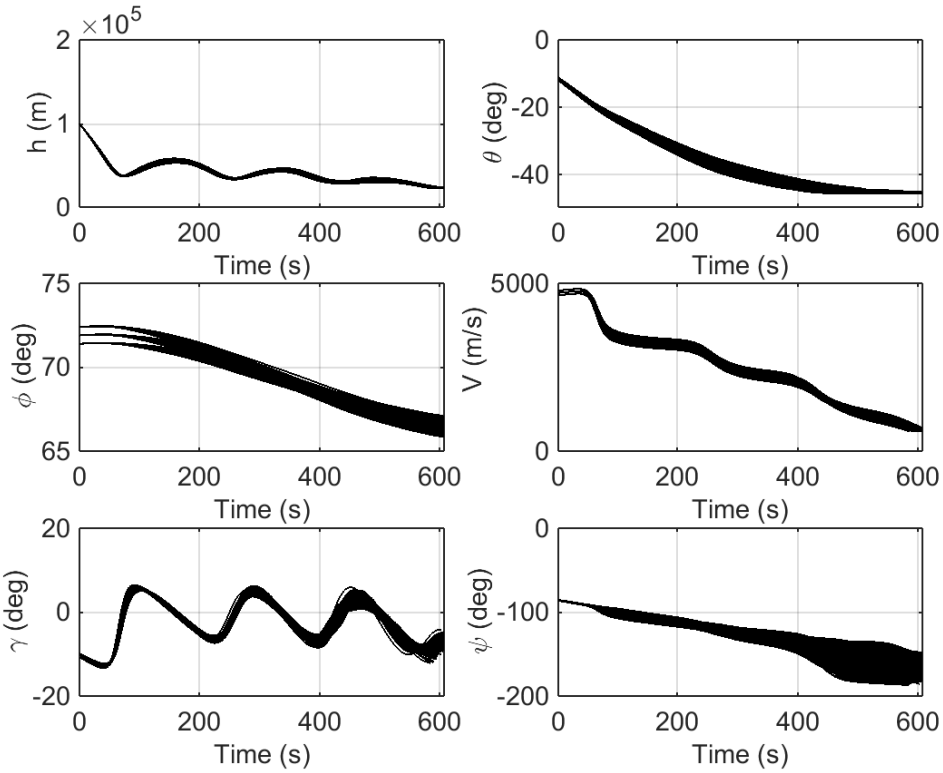


Figure 8.12: Trajectory database - states.

In Fig. 8.14 one can see that the trajectories terminate all in the proximity of the TAEM. The circles show the parametrized dispersions for the latitude and longitude. In 3-D also the altitude parametrization would be seen, while the other 3 uncertainties cannot be visualized, but are taken into account, as one can see from the analysis of the single states. Small variations in the latitude and longitude were observed, but always within the limits defined by the requirements. From Figs. 8.12 and 8.13 one can see that all the states and the controls are smooth, as expected. No violations in terms of constraints (Fig. 8.15) were observed. It is now possible to introduce the algorithms, which will use the generated trajectory database.

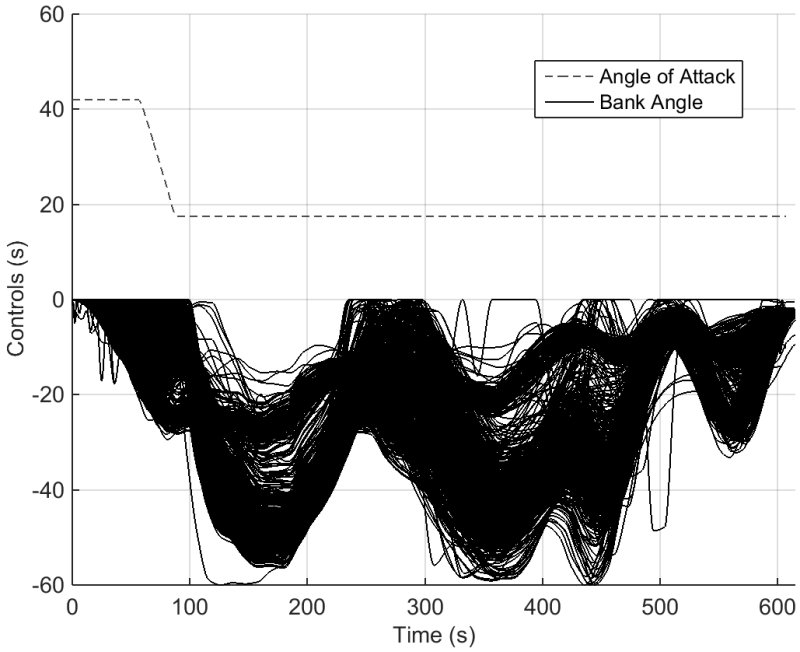


Figure 8.13: Trajectory database - controls.

8.4 Reference Subspace Selection

The d-dimensional space, $\mathcal{P}_c$ previously defined represents all the possible parameters that can differ from their nominal values, and can be estimated (i.e., with the navigation subsystem). In the basic MPI approach[17], for each parameter an upper and a lower value are considered, and a corresponding optimal trajectory is generated. The database of 3^6 extremal trajectories are then taken and combined to provide an approximated solution corresponding to the inflight initial conditions via MPI. In the AMPI, instead of using the entire envelope, only a subspace is selected, and used for the computation of the feedforward guidance solution.

Consistent with what was presented in Section 8, let us consider a 2-D example. Suppose we have the function (in the most generic case this is a vector function) of two variables $\mathbf{F}(X_1, X_2)$, which is known in the points (X_1^L, X_2^L), (X_1^U, X_2^L), (X_1^L, X_2^U), (X_1^U, X_2^U). We are interested to find an approximation of $\mathbf{F}(X_1^i, X_2^i)$ for every couple X_1^i, X_2^i such that $X_1^i \in (X_1^L, X_1^U)$, $X_2^i \in (X_2^L, X_2^U)$. The four extremal pairs defined above represent our trajectory database. The domain appears as plotted in Fig. 8.16.

It is possible to build an approximation of $\mathbf{F}(X_1^i, X_2^i)$ with a multivariate interpolation process (in this case bivariate); if we look at the multivariate process as a black box[1],

[1]The multivariate interpolation is done very efficiently with the Tensor Product Spline, and the DeBoors algorithm. The AMPI deals with how and where this algorithm needs to be evaluated, and for

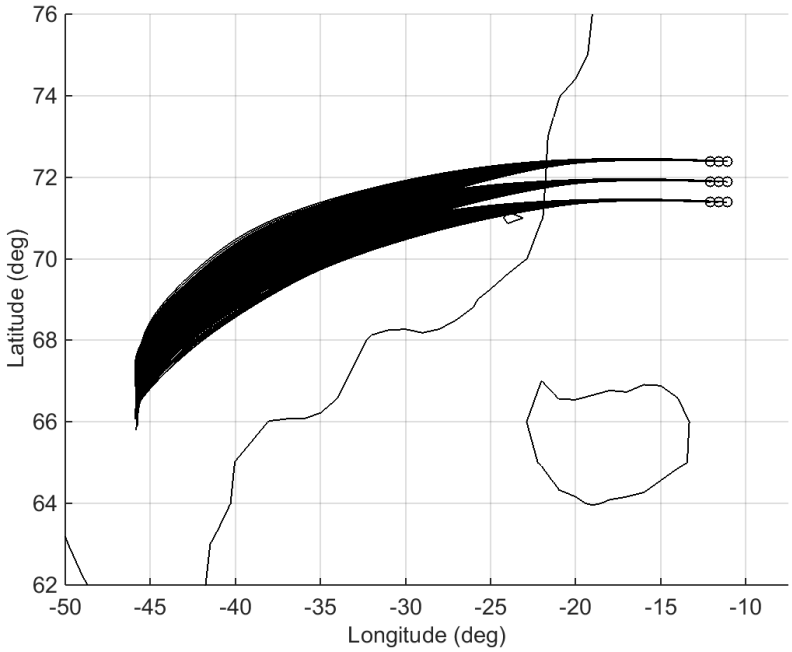


Figure 8.14: Trajectory database - groundtracks.

we can state that

$$\mathbf{F}(X_1^i, X_2^i) \cong \mathbf{MPI}\left[\mathbf{F}(X_1^L, X_2^L), \mathbf{F}(X_1^U, X_2^L), \mathbf{F}(X_1^L, X_2^U), \mathbf{F}(X_1^U, X_2^U)\right] \tag{8.13}$$

The solution is then a function of the information stored in these four points, as shown in Fig. 8.16. The closer the boundary points are, the more accurate the approximation $\mathbf{F}(X_1^i, X_2^i)$ is. For large dispersions this approach may not be accurate enough. In the AMPI the parameter space is organized into a finer grid; then, for each case, the subspace enclosing the point we are interested in is detected, as shown in Fig. 8.17. The approximation will be then built as

$$\mathbf{F}(X_1^i, X_2^i) \cong \mathbf{MPI}\left[\mathbf{F}(X_1^j, X_2^k), \mathbf{F}(X_1^{j+1}, X_2^k), \mathbf{F}(X_1^j, X_2^{k+1}), \mathbf{F}(X_1^{j+1}, X_2^{k+1})\right] \tag{8.14}$$

The objective of this phase is therefore the detection of the indices representing the subspace to use for the synthesis of the feedforward solution. The selection of the reference subspace can be done with algorithm 1, which can be applied to the parameters space $\mathcal{P}$ and to the parametric space p^i, and to the initial conditions x_i, representing the 6-dimensional off-nominal initial conditions. The result will be the six couples of the indices, which detect the subspace to be used for the computation of the adaptive feedforward guidance.

With the application of the algorithm 1, the subspace of the database can be easily

this reason, in this context, it is seen as a black box.

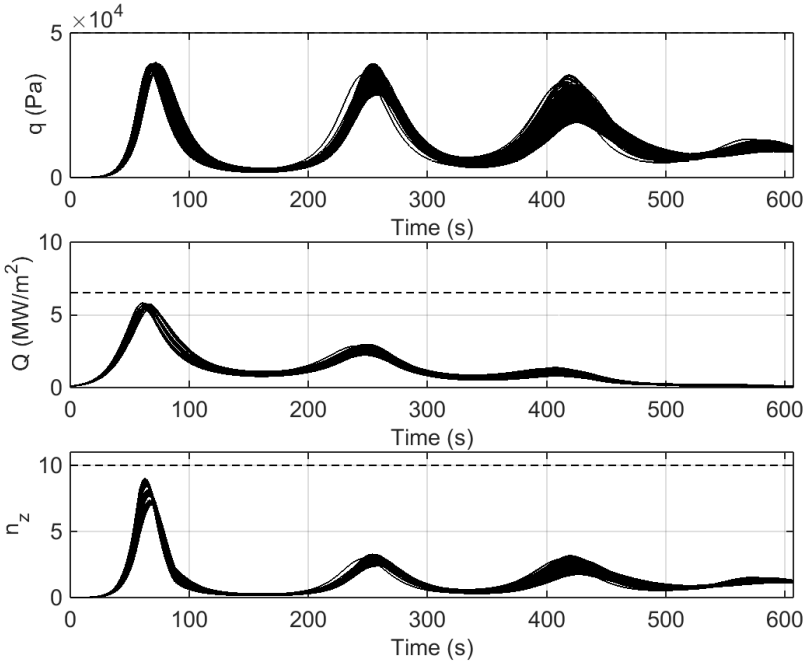


Figure 8.15: Trajectory database - constraints.

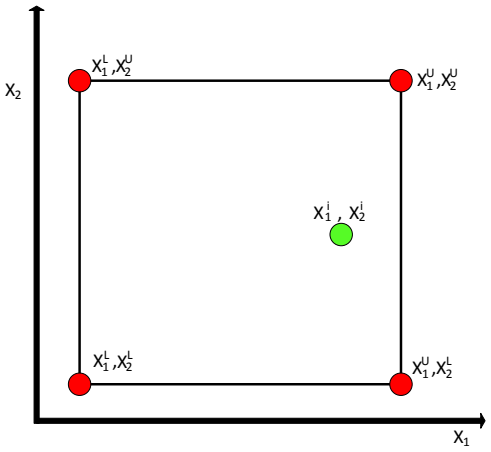


Figure 8.16: Two-dimensional domain example for the application of the multivariate pseudospectral interpolation method.

determined, and it is now possible to compute the feedforward guidance solution via MPI. The drawback of this approach is the rapid increase of memory requirements, since the database increases rapidly. This drawback is balanced by the use of the MPI

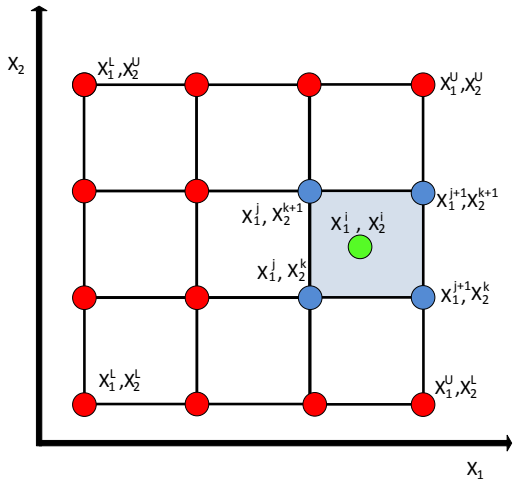


Figure 8.17: Two-Dimensional domain example for the application of the adaptive multivariate pseudospectral interpolation method.

Data: Given: parameter space elements $\mathbf{p}^i$, and initial conditions x_i, $i = 1,\ldots,6$
for i $= 1 : d$
  $\mathbf{eP}^i = \text{sign}(\mathbf{p^i} - \mathbf{U}x_i)$
  for j $= 1 : n_i - 1$
    $\mathbf{\Delta p}^i_j = \mathbf{eP}^i_j\text{-}\mathbf{eP}^i_{j-1}$, $j = 2\ldots,n_i$
    if $\sum_{j=1}^{n_i} \mathbf{eP}^i_j$==-$n_i$
      $[\tilde{h}^i, \tilde{h}^i + 1] = [n_i - 1, n_i]$;
    elseif $\sum_{j=1}^{n_i} \mathbf{eP}^i_j == n_i$
      $[\tilde{h}^i, \tilde{h}^i + 1]$ = [1,2];
    else
      $idx = \mathbf{find}(\mathbf{\Delta p}^i_j \neq 0)$
      $[\tilde{h}^i, \tilde{h}^i + 1] = [idx, idx + 1]$;
    end
  end
Result: 2^d indeces $[\tilde{h}^i, \tilde{h}^i + 1]$, $i = 1\ldots,d$

Algorithm 1: Selection of reference subspace

scheme, which allows to strongly reduce the amount of data needed without losing accuracy, as intuitively explained in Sec. 8.1. How this is done will be explained in the next sections. In the parametric six-dimensional space, each subspace selected for the synthesis of the guidance solution is made of 2^6 neighboring trajectories, combined to provide an approximation of the functions $\mathbf{F}$, which, in our case, include the states, as

well as the bank angle, and the final time. This interpolated solution will represent the feedforward guidance, which can be adapted to any initial condition compatible with the extension of the database. Once the subspace, represented by the closest 2^d trajectories to the off-nominal initial conditions, is determined, it is possible to apply the MPI as explained in the next sections.

8.5 Low-Density Multivariate Interpolation

As stated in the previous section, a solution of the OCP previously defined depends not only on the specific choice of a parameter vector $\mathbf{p} \in \mathcal{P}_c$, but also on time. Within this work the computation of a solution of an OCP is based on the solution of a corresponding NLP. The interpolation approach described in this section shows how to compute a given point of an interpolated trajectory by using the corresponding values of the trajectory-database. Since we are using a transcription into a NLP based on the flipped Radau pseudospectral (FRP) method [16,17], we need to perform this interpolation at each of the collocation nodes defined by the roots of the FRP. These nodes (and their related polynomials) are properly defined in Chapter 7, and are given by Eqs. (7.11) and (7.13). The set of all collocation nodes can be seen as the domain of the low-density discrete solution.

Once the reference subspace has been selected, we have the d pairs of parameters $\mathbf{p} \in \mathcal{P}_c$, which indicate what specific trajectories of our database will be involved in the multivariate interpolation. As in the 1-D example of Sec. 8.1 we had one pair of values of $\mathbf{p}$, (in this case x_0, which is 0.5 and 1.5) and 2^1 trajectories for the computation of the interpolated trajectory. In the most general case, we have 2^d trajectories for a $d-$dimensional parameter $\mathbf{p}$. This means that at each timestep, 2^d values for each of the states and controls will be used for the multivariate interpolation process.

Let us define the generic variable f (i.e., a state or a control) of the optimal trajectories set, associated with the 2^d values of $\mathbf{p}$ determined by the reference subspace algorithm. To make the notation simple, let us suppose that the reference subspace identifies the indices 1 and 2 for each of the d components of $\mathbf{p}$. For a d-dimensional off-nominal value of x_i, $i = 1, \cdots, d$, we have

$$x_i \in [p_1^i,\ p_2^i],\ i = 1, \cdots, d \tag{8.15}$$

which means that, as in the example of Sec. 8.1, the off-nominal vector has all its components defined in the intervals determined by the parameter subspace. At each timestep τ_k, and for each of the variables included in the trajectories we have 2^d values of $f(\tau_k, \mathbf{p}_i)$, belonging to the database, which need to be interpolated to provide an intermediate value $f_{int}(\tau_k, x_i)$ consistent with the d off-nominal initial conditions x_i. We can indicate the 2^d values of $f(\tau_k, \mathbf{p}_i)$ as $\mathbf{F}(\tau_k, \mathbf{p}_i)$. The grid points $\mathbf{p}_i$ are called supporting points while the $\mathbf{F}(\tau_k, \mathbf{p}_i)$ are called supporting values. With these definitions it is now possible to define the tensor product spline. A tensor product spline

$s \in \mathbb{S}_{k_1,\mathbf{t}_1} \bigotimes \cdots \bigotimes S_{k_d,\mathbf{t}_d}$ on a grid $\mathbf{P}$ at a grid point $\mathbf{x} \in \mathbf{P}$ is in general defined by

$$s(\mathbf{x}) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_d=1}^{m_d} c_{i_1,\ldots,i_d} B_{i_1,k_1}(x_1) \ldots B_{i_d,k_d}(x_d). \tag{8.16}$$

In Eq. (8.16) $B_{i,k}$ denotes the i^{th} B-spline of order k for a given non-decreasing knot vector $\mathbf{t} = (t_i)_{i=1}^{m+k}$. The coefficients

$$\mathbf{C} = (c_{i_1,\ldots,\imath_d})_{i_1=1,\ldots,i_d=1}^{m_1\ ,\ldots,m_d} \tag{8.17}$$

are computed, such that the resulting tensor product spline $s(\mathbf{x})$ fulfills the interpolation condition, that is

$$s(\mathbf{p_i}) = f(\tau_k, \mathbf{p}_i) \,\forall\, i \in 1 \cdots d,\ k = 1, \cdots, N_{LD} \tag{8.18}$$

In our case, the interpolated variable $f_{int}(\tau_k, x_i)$ will be

$$f_{int}(\tau_k, \mathbf{x}_i) = s(\mathbf{x}_i) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_d=1}^{m_d} c_{i_1,\ldots,i_d} B_{i_1,k_1}(x_1) \ldots B_{i_d,k_d}(x_d). \tag{8.19}$$

Since every B-spline in Eq. (8.16) depends only on a single variable, the d-variate interpolation problem can be divided into d univariate problems. Each univariate spline interpolation is solved via the numerical stable and efficient algorithm of De Boor [68]. Depending on the choice of k and a suitable knot vector, $\mathbf{t}$, the tensor product spline interpolation in Eq. (8.16) corresponds to an interpolation method such as piecewise constant, piecewise linear or piecewise cubic Hermite interpolation.

In general, the tensor-product spline interpolation allows the use of different interpolation methods for each of the d dimensions. It is now necessary to determine the knot vectors $\mathbf{t}_i$ and the coefficient matrices $\mathbf{C}$. Within this work the grid $\mathbf{P}$ consists of two supporting points for each direction, such that $n_i = 2, i = 1, \ldots, d$ and

$$\mathbf{P} = \left\{p_1^1, p_2^1\right\} \times \cdots \times \left\{p_1^d, p_2^d\right\}. \tag{8.20}$$

The choice of the order of the splines $k_i = 2, i = 1, \ldots, d$ and the corresponding choice of the knot vectors

$$\mathbf{t}_i = \left(t_j^i\right)_{j=1}^4 = \left(p_1^i, p_1^i, p_2^i, p_2^i\right), \qquad i = 1, \ldots, d \tag{8.21}$$

defines a piecewise linear interpolation in each direction on the given d-dimensional grid. For $\mathbf{C} = \mathbf{F}(\tau_k, \mathbf{p}_i)$ the piecewise linear interpolation method fulfills the interpolation condition given by Eq. (8.18) and there is no additional computational effort required to determine the coefficients $\mathbf{C}$. With these simplifications, the evaluation of the tensor-product spline interpolation at a given point of the parameter space $\mathbf{p} \in \mathbf{P}_c$ is described by the pseudo code according to algorithm 2. The EvalUnivSpline$(\mathbf{t}, \mathbf{c}, x_i)$ function in algorithm 2 denotes the evaluation of a univariate spline with coefficients $\mathbf{C}$ at a point

Data: Given: knot vector $\mathbf{t}$, coefficients $\mathbf{C}$, spline $s \in \mathbb{S}_{2,\mathbf{t}_1} \bigotimes \cdots \bigotimes \mathbb{S}_{2,\mathbf{t}_d}$, evaluation point $\mathbf{x}_i \in \mathbf{P}_c$
$\mathbf{A}_0 = \mathbf{C}$;
for i = 1 : d
  $\mathbf{A}_i = \text{EvalUnivSpline}(\mathbf{t}_i, \mathbf{A}_{i-1}, x_i)$;
  $\mathbf{A}_i = \mathbf{A}_i'$;
end
$s(\mathbf{p}) = \mathbf{A}_d$;
Result: interpolated values $s(\mathbf{x}_i) = f(\tau_k, \mathbf{x}_i)$

Algorithm 2: Tensor Product Spline Interpolation

x_i while the operator $(\cdot)'$ performs a cyclic rotation, such that $\mathbf{A} \in \mathbb{R}^{n_1,n_2,\ldots,n_d} \Rightarrow \mathbf{A}' \in \mathbb{R}^{n_2,\ldots,n_d,n_1}$.

Since this process is applied to all the states and the controls belonging to the trajectory, the interpolated solution will be represented by the $N_{LD}+1$ values of each of the states, that is, the matrix $\mathbf{X}_{LD}$, having dimensions $n_s \times N_{LD}$, where n_s is the number of states associated with our problem, and the controls $\mathbf{U}_{LD}$, having dimensions $n_c \times N_{LD}$, where n_c is the number of controls of our formal OCP. The generated solution will have initial states corresponding to the off-nominal d-dimensional initial condition, and therefore the generated trajectory will null the initial error introduced by off-nominal conditions.

This process delivers us the formal low-density discrete solution. It is important to emphasize that interpolated reference trajectories in general are not formally solutions of the equations of motion that are used within the underlying OCP. However, they represent a good, real-time capable, approximation of the optimal solutions without the computational burden needed to generate them. Let us briefly remind the following properties (already introduced in Chapter 7), that, together with the ease of implementation, justify the choice of using the pseudospectral methods for the characterization of the discrete domain.

- Spectral convergence in the case of a smooth problem;
- Straightforward implementation;
- Sparse structure of the associated NLP problem;
- Mapping between the costates of the NLP discrete solution and the costates of the optimal continuous solution in virtue of the pseudospectral covector mapping theorem[60];
- Removal of the Runge phenomenon[47,59];

The removal of the Runge phenomenon has an important implication: since all the polynomials generated using the FRP nodes do not have undesired oscillations, the

interpolated solutions computed in these points will be smooth as well. Therefore, database storing accurate trajectories can be reduced to storing the nodal values, which can be converted into high-density discrete solution with no need to evaluate splines, as we will see in the next Section. This approach significantly reduces the on-board memory requirements, as well as the on-board CPU burden.

8.6 Computation of Pseudospectral-based high-density discrete solutions

Let us suppose to have computed the values representing the LD discrete solutions in the $N_{LD}+1$ FRP nodes (that is, the N_{LD} FRP nodes plus the node at -1. The solution is formed by the time vector $\mathbf{t_{LD}}$, the states $\mathbf{X_{LD}}$, and by the controls $\mathbf{U}_{LD}$. The matrices $\mathbf{X}_{LD}$ and $\mathbf{U}_{LD}$ have dimensions $n_s \times (N_{LD}+1)$ and $n_c \times (N_{LD}+1)$, respectively, where n_s and n_c are the number of states and controls associated with the problem under analysis. We can group the states and the controls in a matrix $\mathbf{T}_{LD}$, having dimensions $(n_s+n_c) \times (N_{LD}+1)$.

$$\mathbf{T}_{LD} = \left\{ \begin{array}{c} \mathbf{X}_{LD} \\ \mathbf{U}_{LD} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_{N_{LD}} \\ \mathbf{U}_0, \mathbf{U}_1, ..., \mathbf{U}_{N_{LD}} \end{array} \right\} \tag{8.22}$$

Our objective is to efficiently convert the matrix $\mathbf{T}_{LD}$ into a matrix $\tilde{\mathbf{T}}_{HD}$ representing the HD discrete solution,

$$\tilde{\mathbf{T}}_{HD} = \left\{ \begin{array}{c} \tilde{\mathbf{X}}_{HD} \\ \tilde{\mathbf{U}}_{HD} \end{array} \right\} = \left\{ \begin{array}{c} \tilde{\mathbf{X}}_0, \tilde{\mathbf{X}}_1, ..., \tilde{\mathbf{X}}_{N_{HD}} \\ \tilde{\mathbf{U}}_0, \tilde{\mathbf{U}}_1, ..., \tilde{\mathbf{U}}_{N_{HD}} \end{array} \right\} \tag{8.23}$$

where $N_{HD}+1$ is the number of points representing the HD discrete solution. Moreover, the HD time vector $\tilde{t}_{HD}$ must be computed. If we use the definition in Eq. (7.11), we can write

$$F(\tau) = \sum_{i=0}^{N_{LD}} F_i \prod_{\substack{k=0 \\ k \neq i}}^{N_{LD}} \frac{\tau - \tau_k}{\tau_i - \tau_k},\ \tau \in [-1, 1] \tag{8.24}$$

where F_i represents a low-density variable. It can be replaced with the i^{th} element of the p^{th} row of $\mathbf{T}_{LD}$. Moreover, the continuous variable $\tau \in [-1, 1]$ can be sampled in the $N_{HD}+1$ high-density discrete nodes. The result will be the high-density representation of our variables

$$\tilde{T}^p_{HD}(\tilde{\tau}_m) = \sum_{i=0}^{N} T^p_{LD,i} \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{\tilde{\tau}_m - \tau_k}{\tau_i - \tau_k},\ p = 1, ...(n_s+n_c),\ m = 0, ...(N_{HD}) \tag{8.25}$$

Expression (8.25) can be extended to all the rows of the matrix $\tilde{\mathbf{T}}_{HD}$, and rewritten in matrix form as

$$\tilde{\mathbf{T}}_{HD} = \mathbf{T}_{LD}\mathbf{P}_{FRP} \tag{8.26}$$

where the matrix $\mathbf{P}_{FRP}$ has dimensions $(N_{LD}+1)\times(N_{HD}+1)$, and is given by

$$\mathbf{P}_{FRP}=\begin{bmatrix} \prod_{k=1}^{N_{LD}} \frac{\tilde{\tau}_0-\tau_k}{\tau_0-\tau_k} & \dots & \prod_{k=1}^{N_{LD}} \frac{\tilde{\tau}_{N_{HD}}-\tau_k}{\tau_0-\tau_k} \\ \dots & \dots & \dots \\ \prod_{k=0}^{N_{LD}-1} \frac{\tilde{\tau}_0-\tau_k}{\tau_{N_{LD}}-\tau_k} & \dots & \prod_{k=0}^{N_{LD}-1} \frac{\tilde{\tau}_{N_{HD}}-\tau_k}{\tau_{N_{LD}}-\tau_k} \end{bmatrix} \tag{8.27}$$

The elements $\tilde{\tau}_m$ represent the high-density discrete pseudotime vector, defined between -1 and 1. Since both the nodes τ_k, where the solutions are computed, and the nodes $\tilde{\tau}_m$, where the solutions are effectively evaluated, are part of the process of the database generation (as they are part of the transcription), the matrix $\mathbf{P}_{FRP}$ can be computed offline and stored, with a significant saving in CPU time, and the trajectory synthesis is reduced to a multivariate linear interpolation process and to the matrix multiplication defined in Eq. (8.26). To complete the generation of the HD solution, we still need the HD discrete physical time vector associated to the interpolated solution. It can be computed by using the following expression:

$$\tilde{t}_m=\frac{\tilde{t}_f-\tilde{t}_0}{2}\tilde{\tau}_m+\frac{\tilde{t}_f+\tilde{t}_0}{2},\ \ m=0,...,N_{HD} \tag{8.28}$$

The initial time $\tilde{t}_0$ is given by the initial time t_0. The final time $\tilde{t}_f$ is computed by applying the multivariate interpolation approach described in algorithms 1 and 2 to the final times stored in the trajectory database. The trajectory representing the feedforward guidance solution is completely generated with the application of the algorithms 1 and 2, and Eqs. (8.26) and (8.28).

8.7 Numerical Examples

In this Section some numerical examples are provided. Specifically, to show the improvements obtained with the adaptive feedforward guidance, 100 cases have been run and compared with the results obtained by tracking the nominal reference solution associated with the condition $\mathbf{x}(t_0)=\mathbf{x}^*(t_0)$. The nature of the feedback controller will be clarified in Chapter 9, and is not relevant here, as it is exactly the same for the two guidance systems.

From the analysis of Figs. 8.18, 8.19, and 8.20 we can see that all the states and the controls satisfy the limits listed in Tables 2.2 and 2.3. No constraints violations have been observed (Fig. 8.21). Both the longitudinal and the lateral performances are improved. Indeed, Fig. 8.22 and 8.23 show the improvements in terms of final range-to-go and crossrange dispersions. State errors are plotted in Fig. 8.24, while histograms Figs. 8.25 and 8.26 report the final dispersions for the longitudinal states, that is, the final altitude, and the final velocity. Indeed, the introduction of the adaptive feedforward guidance scheme allows the system to reduce the error w.r.t. the nominal interfaces of 23.5 km and 595 m/s. Finally, we can observe the improvement of the final range-to-go, defined as the missing distance w.r.t. the nominal terminal interface.

In general we can state that the AMPI allows to strongly improve the performance. Indeed, with the current setup, 91% of the cases fall within a radius of 5 km w.r.t. the nominal solution even in presence of initial dispersion in the order of 100 km in terms of latitude and longitude.

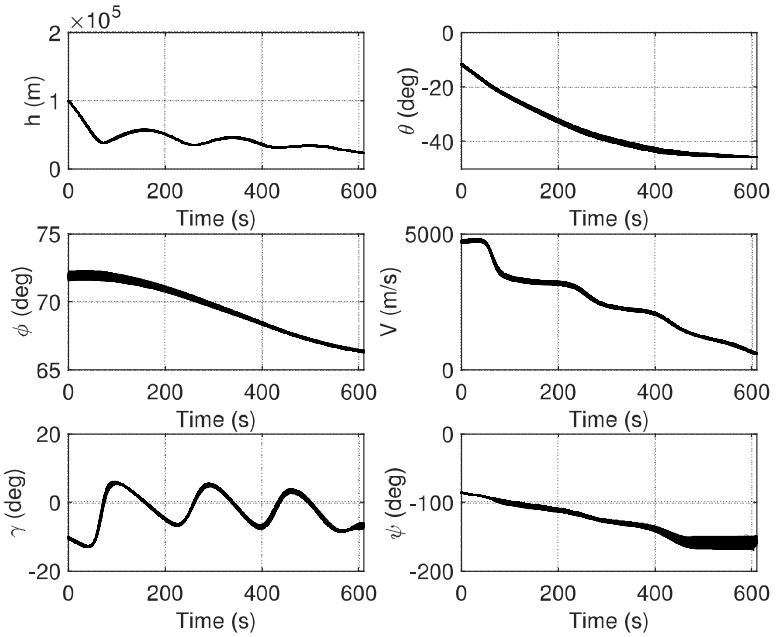


Figure 8.18: Performance of the adaptive feedforward guidance - states.

It is possible to observe how, even in presence of a strong asymmetric scenario, and with large initial dispersions, this approach leads to reduce the error on the range-to-go to less than 5 km w.r.t. the prescribed terminal position for more than 90% of the cases. However, as previously stated, other uncertainties play a role during the entry. Therefore, their effect must be limited by properly formulating a feedback controller. This is the subject of the Chapter 9.

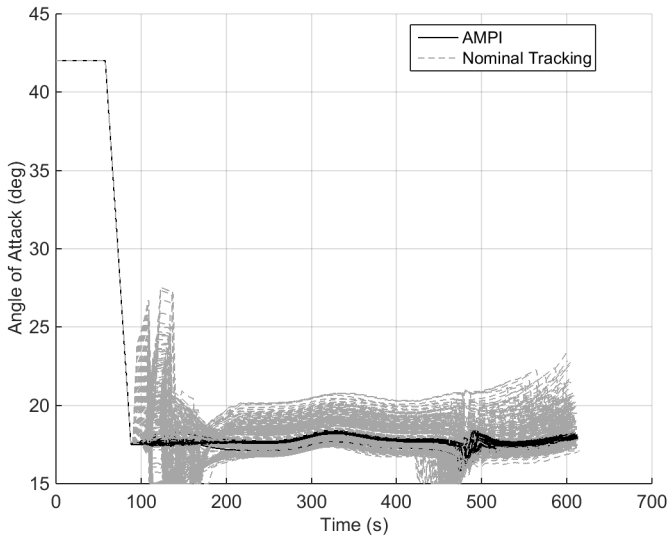


Figure 8.19: Performance of the adaptive feedforward guidance - angle of attack.

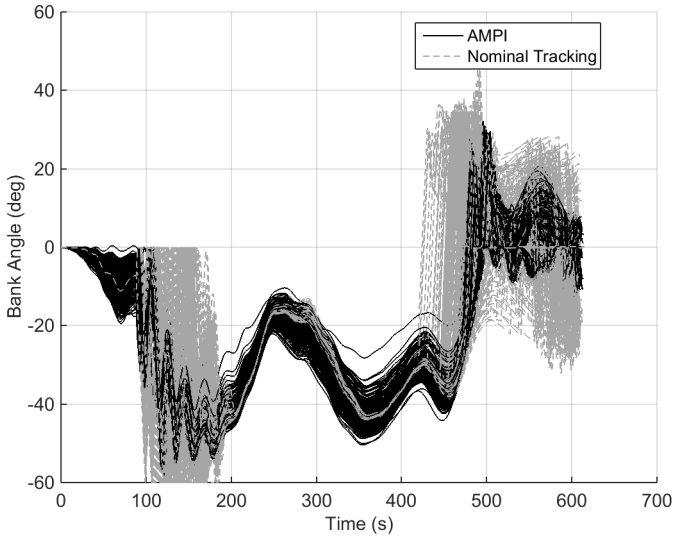


Figure 8.20: Performance of the adaptive feedforward guidance - bank angle.

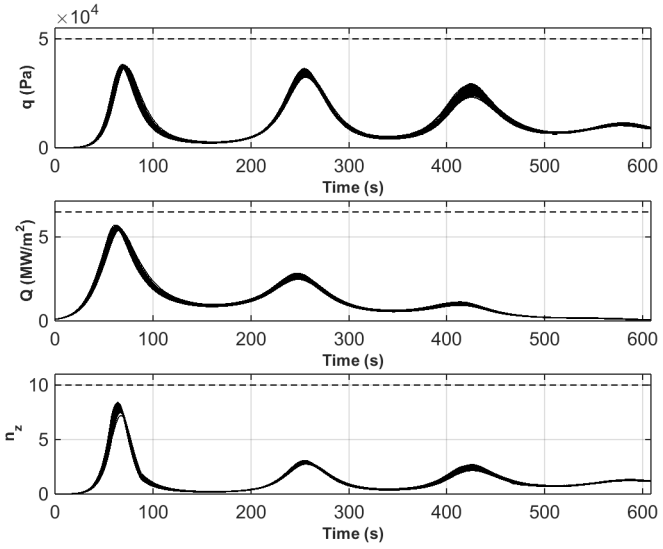


Figure 8.21: Performance of the adaptive feedforward guidance - constraints.

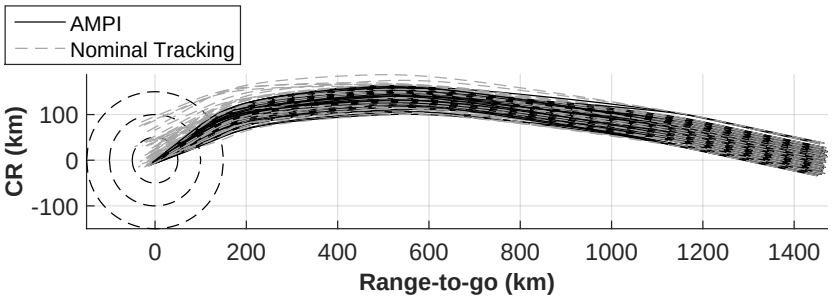


Figure 8.22: Performance of the adaptive feedforward guidance - groundtracks.

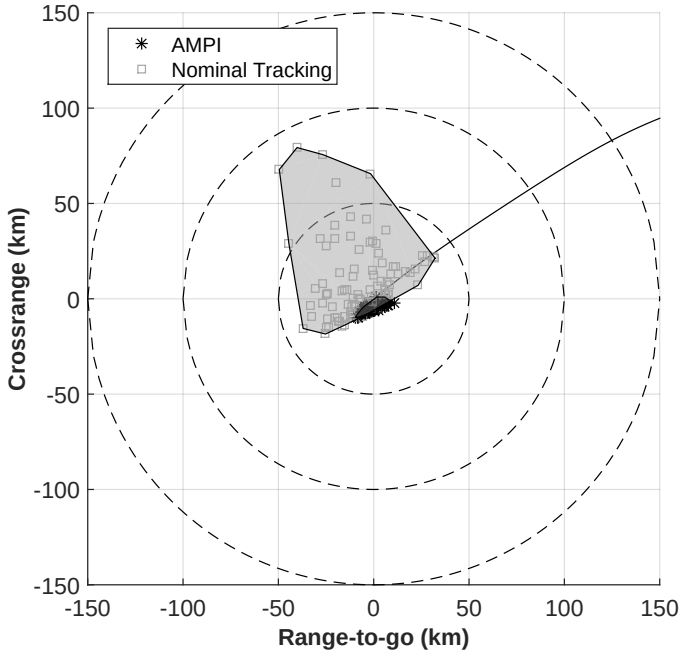


Figure 8.23: Performance of the adaptive feedforward guidance - footprints.

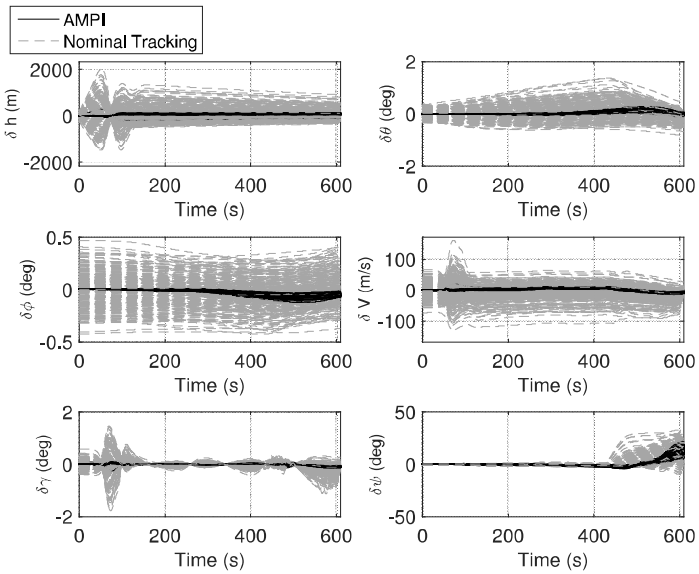


Figure 8.24: Performance of the adaptive feedforward guidance - state errors.

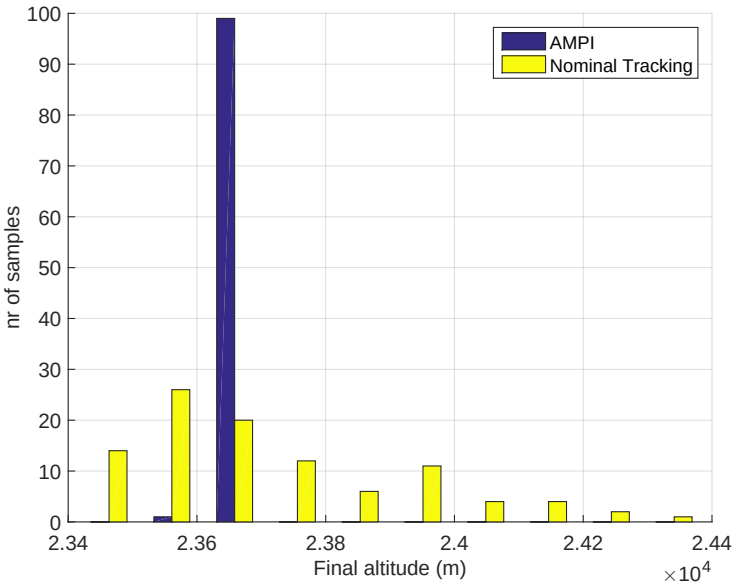


Figure 8.25: Performance of the adaptive feedforward guidance - final-altitude dispersions.

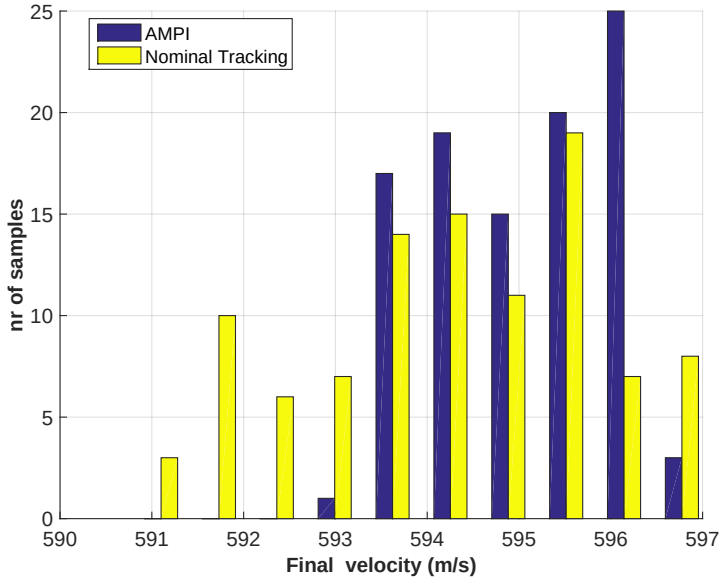


Figure 8.26: Performance of the adaptive feedforward guidance - final-velocity dispersions.

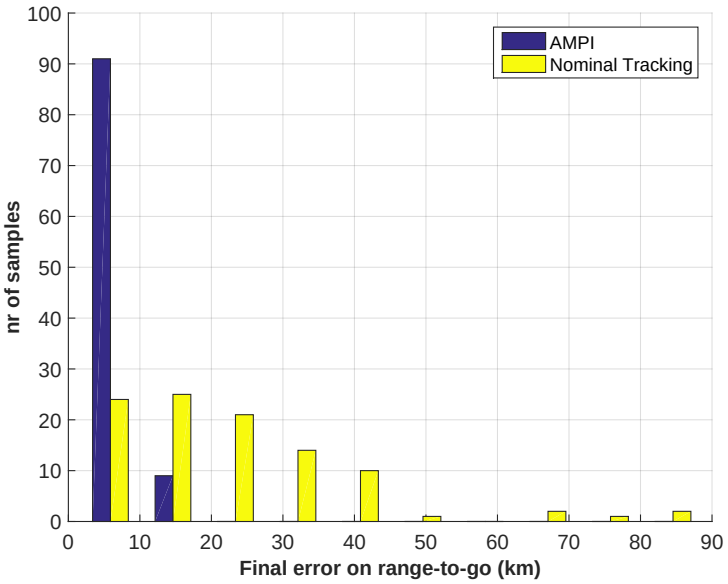


Figure 8.27: Performance of the adaptive feedforward guidance - final range-to-go dispersions.

Chapter 9

Robust Feedback Guidance

The main objective of this chapter is to develop the feedback controller which will reduce the error coming from uncertainties and disturbances acting on the system, in the limits of the vehicle's capabilities. The feedback controller is based on sliding mode control. In Sec. 9.1 the concept of input/output linearization (also known as feedback linearization) will be formally introduced. In Sec. 9.2.1 this technique will be applied to Eq. (4.12) to obtain the corresponding MIMO feedback-linearized system. After the formal introduction of the sliding surfaces (Sec. 9.2.2), the design of the sliding mode control associated with such system is the focus of Sec. 9.2.3, while in Sec. 9.2.4 details about applicability, stability, and implementation will be given. However, the longitudinal controller designed via SMC is not sufficient as stand-alone, as it does not take the heading error into account. Therefore, in Sec. 9.3 the lateral control, which is integrated within the longitudinal tracking controller, is described. Finally, Sec. 9.4 shows the results obtained by using the SMC controller in nominal conditions.

9.1 Sliding Mode Theory - Input Output Linearization

Sliding mode control (SMC) or variable structure control (VSC) is a powerful class of methods to control nonlinear systems[69,70,71]. It is based on the concept of a *sliding surface*, defined by the equation

$$\sigma(x) = 0 \tag{9.1}$$

where $\sigma(x)$ is the sliding variable, function of the scalar state x of the system to be controlled. The idea is to define the control law so that the system under investigation behaves in a desired way, and to find the conditions, which realize this behavior. An extensive explanation of sliding mode control can be found in several sources[54,72,73], where the main principles are properly summarized. The use of SMC relies on the possibility to define the nonlinear system in the so-called input/output linearized form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) + \sum_{k=1}^{n} \mathbf{g_k}(\mathbf{x})\mathbf{u_k}\mathbf{y_i}(t) \quad = \mathbf{h_i}(\mathbf{x}),\ i = 1, \cdots, m \tag{9.2}$$

$\mathbf{f}$, $\mathbf{g_k}$ and $\mathbf{h_i}$ are smooth functions of the state $\mathbf{x} \in \Re^n$, representing the dynamics, the control dependency, and the output of the system we want to control. The SMC theory gives the possibility to design a control able to track a given output $\mathbf{y_i(t)}$. This is done by building a relationship between the $r_i{}^{\text{th}}$ derivative of the output we want to track, and the controls. If we differentiate the second of Eq. (9.2) r_i times, and we take the first of Eq. (9.2) into account, one can see that

$$\mathbf{y_i^{r_i}(t)} = \mathbf{L_f^{r_i}(h_i)} + \sum_{k=1}^{m} \mathbf{L_{g_k}} \left(\mathbf{L_f^{r_i - 1}(h_i)} \right) \mathbf{u_k} \tag{9.3}$$

holds. The operator $\mathbf{L_f^{r_i}(h_i)}$ is the Lie derivatives of order r_i of function $\mathbf{h_i}$ w.r.t. $\mathbf{f}$[74]. Lie Derivative (or directional derivative) of the function $\mathbf{h_i} \in \Re$ with respect to a function $\mathbf{f} \in \Re^n$ is defined as

$$\mathbf{L_f(h_i)} = \nabla \mathbf{h_i} \cdot \mathbf{f(x)} \tag{9.4}$$

and

$$\nabla = \left[\begin{array}{ccc} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{array} \right] \tag{9.5}$$

Derivatives of higher order can be recursively computed as

$$\mathbf{L_f^r(h_i)} = \mathbf{L_f} \left(\mathbf{L_f^{r-1}(h_i)} \right) \tag{9.6}$$

and this definition can be applied to obtain the r_i^{th} derivative of the i^{th} output $\mathbf{y_i}(t)$ to track. Moreover, we define $\mathbf{r} = \sum \mathbf{r_i}$ relative degree of the system. If the relative degree is equal to the sum of the orders of the differential equations considered, it is possible to express a generic system of nonlinear differential equations into an equivalent form given by Eq. (9.2), where the controls are separated, and appear as linear combinations weighted by the functions $\mathbf{g_k(x)}$, (i.e., the system is in an affine form). From the practical point of view, to obtain the form of Eq. (9.2), each differential equation involving the variable to track must be derived until the controls appear linearly. After having checked that the relative degree of the system is equal to the sum of orders of the initial differential equations, it is possible to apply the SMC.

9.2 Longitudinal Controller

9.2.1 Input/Output form of Equations of motion

Let us apply the approach reported in the previous section to the longitudinal equations of motion. According to the model represented by Eq. (4.12), the longitudinal dynamics of the vehicle, in absence of Earth's rotation, is described as

$$\begin{aligned} \dot{h} &= V \sin\gamma \\ \dot{V} &= -D - g \sin\gamma \\ \dot{\gamma} &= \frac{1}{V} L \cos\sigma + \left(\frac{V}{r} - \frac{g}{V} \right) \cos\gamma \end{aligned} \tag{9.7}$$

We can add two artificial states, which are the angle-of-attack and the bank-angle:

$$\begin{aligned}\dot{\alpha} &= u_\alpha + \dot{\alpha}_{ref} \\ \dot{\sigma} &= u_\sigma + \dot{\sigma}_{ref}\end{aligned} \tag{9.8}$$

The objective is to derive the nonlinear sliding mode control that allows to track the reference altitude $h_{ref}(t)$ and speed $V_{ref}(t)$. An important difference w.r.t. other SMC-based techniques[75,76,77] is that in this case the reference states are time-dependent, therefore their derivatives are different from zero, and need to be included in the controller design.

As stated, we want our controls (in this case the angle of attack rate $\dot{u}_\alpha$, and the bank angle rate $\dot{u}_\sigma$) to appear linearly in the equations of motion. This happens if we differentiate the equation of the altitude three times, and the equation of the velocity two times. Then, the total relative degree of the system is equal to 5, and is equal to the order of the system associated with Eqs. (9.7) and (9.8). If we derive the altitude w.r.t. the time three times, we get

$$\begin{aligned}\dot{h}(t) &= V\sin\gamma \\ \ddot{h}(t) &= \dot{V}\sin\gamma + V\dot{\gamma}\cos\gamma \\ \dddot{h}(t) &= \ddot{V}\sin\gamma + 2\dot{V}\dot{\gamma}\cos\gamma + V\ddot{\gamma}\cos\gamma - V\dot{\gamma}^2\sin\gamma\end{aligned} \tag{9.9}$$

It is possible to see that the expression for $\ddot{\gamma}$ is needed. If we derive γ twice, we get

$$\begin{aligned}\dot{\gamma} &= \frac{L}{V}\cos\sigma + \left(\frac{V}{r} - \frac{g}{V}\right)\cos\gamma \\ \ddot{\gamma} &= \frac{\dot{L}}{V}\cos\sigma - \frac{L\dot{V}}{V^2}\cos\sigma - \frac{L\sin\sigma}{V}(\dot{\sigma}_{ref} + u_\sigma) + \cdots \\ &\quad + \left(\frac{\dot{V}}{r} - \frac{V\dot{r}}{r^2} - \frac{\dot{g}}{V} + \frac{g\dot{V}}{V^2}\right)\cos\gamma - \left(\frac{V}{r} - \frac{g}{V}\right)\dot{\gamma}\sin\gamma\end{aligned} \tag{9.10}$$

The second variable we want to track is the velocity modulus V. Therefore, let us differentiate the velocity w.r.t. the time twice. We get

$$\begin{aligned}\dot{V} &= -D - g\sin\gamma \\ \ddot{V} &= -\dot{D} - \dot{g}\sin\gamma - g\dot{\gamma}\cos\gamma\end{aligned} \tag{9.11}$$

Atmospheric density and gravity acceleration derivatives w.r.t. time can be easily computed either analytically or numerically. Assuming that the atmospheric density ρ and the gravity acceleration g depend only on altitude, we can write

$$\begin{aligned}\dot{\rho} &= \rho_h\dot{h} \\ \dot{g} &= g_h\dot{h}\end{aligned} \tag{9.12}$$

where ρ_h and g_h are the derivatives of the atmospheric density, and the gravitational acceleration w.r.t. altitude, respectively.

The derivatives of the drag and lift accelerations are

$$\begin{aligned}\dot{D} &= D\left(\frac{\dot{\rho}}{\rho} + 2\frac{\dot{V}}{V} + \frac{\dot{C}_D}{C_D}\right)\\ \dot{L} &= L\left(\frac{\dot{\rho}}{\rho} + 2\frac{\dot{V}}{V} + \frac{\dot{C}_L}{C_L}\right)\end{aligned} \tag{9.13}$$

and the time derivatives of the aerodynamic coefficients can be computed using the information contained in the aerodynamic database,

$$\begin{aligned}\dot{C}_D &= C_{D,\alpha}\left(\dot{\alpha}_{ref} + u_\alpha\right) + C_{D,M}M_V\dot{V} + (C_{D,M}M_h + C_{D,h})\dot{h}\\ \dot{C}_L &= C_{L,\alpha}\left(\dot{\alpha}_{ref} + u_\alpha\right) + C_{L,M}M_V\dot{V} + (C_{L,M}M_h + C_{L,h})\dot{h}\end{aligned} \tag{9.14}$$

where:

- $C_{D,\alpha}$ and $C_{L,\alpha}$ are the derivative of the aerodynamic coefficients w.r.t. the angle of attack α;
- $C_{D,M}$ and $C_{L,M}$ are the derivative of the aerodynamic coefficients w.r.t. the Mach number M;
- $C_{D,h}$ and $C_{L,h}$ are the derivative of the aerodynamic coefficients w.r.t. the altitude;
- $\dot{\alpha}_{ref}$ is the reference angle of attack rate, provided by the feedforward solution, together with the bank angular rate $\dot{\sigma}_{ref}$;
- M_V and M_h are the derivatives of the Mach number w.r.t. the velocity V and the altitude h, equal to $\frac{1}{\sqrt{\gamma_{air}R^*T}}$ and $-\frac{M}{2}\frac{T_h}{T}$, with T_h representing the derivative of the atmospheric temperature T w.r.t. h;

With all these relationships, the entire input/output model can be obtained. It has the following compact form

$$\begin{aligned}\dddot{h}(t) &= a_h + b_{h,\alpha}u_\alpha + b_{h,\sigma}u_\sigma\\ \ddot{V}(t) &= a_v + b_{V,\alpha}u_\alpha + b_{V,\sigma}u_\sigma\end{aligned} \tag{9.15}$$

It is possible to demonstrate that, starting from Eqs. (9.9), (9.10), and (9.11), and combining them with Eqs. (9.12), (9.13), and (9.14), the terms a_h, a_V, $b_{h,\alpha}$, $b_{h,\sigma}$, $b_{V,\alpha}$,

$b_{V,\sigma}$ can be computed as follows:

$$\begin{aligned}
a_h &= 2\dot{V}\dot{\gamma}\cos\gamma - V\dot{\gamma}^2\sin\gamma + \sin\gamma a_V + V\cos\gamma a_\gamma \\
a_V &= -a_D - g_h\sin\gamma\dot{h} - g\cos\gamma\dot{\gamma} \\
b_{h,\alpha} &= \sin\gamma b_{V,\alpha} + V\cos\gamma b_{\gamma,\alpha} \\
b_{h,\sigma} &= V\cos\gamma b_{\gamma,\sigma} \\
b_{V,\alpha} &= -b_{D,\alpha} \\
b_{V,\sigma} &= 0 \\
a_D &= D\left(\frac{\rho_h}{\rho} + \frac{C_{D,h}}{C_D}\right)\dot{h} + D\left(\frac{2}{V} + \frac{C_{D,M}M_V}{C_D}\right)\dot{V} + D\left(\frac{C_{D,\alpha}}{C_D}\right)\dot{\alpha}_{ref} \\
a_L &= L\left(\frac{\rho_h}{\rho} + \frac{C_{L,h}}{C_L}\right)\dot{h} + L\left(\frac{2}{V} + \frac{C_{L,M}M_V}{C_L}\right)\dot{V} + L\left(\frac{C_{L,\alpha}}{C_L}\right)\dot{\alpha}_{ref} \\
b_{D,\alpha} &= D\frac{C_{D,\alpha}}{C_D} \\
b_{L,\alpha} &= L\frac{C_{L,\alpha}}{C_L} \\
a_\gamma &= \left(\frac{1}{r} + \frac{g}{V^2}\right)\dot{V} - \left(\frac{V}{r^2} + \frac{g_h}{V}\right)\dot{h}\cos\gamma + \cdots \\
&\quad - \frac{L}{V^2}\dot{V}\cos\sigma - \frac{L}{V^2}\sin\sigma\dot{\sigma}_{ref} - \dot{\gamma}\left(\frac{V}{r} - \frac{g}{V}\right)\sin\gamma + \frac{a_L}{V}\cos\sigma \\
b_{\gamma,\alpha} &= \frac{b_L}{V}\cos\sigma \\
b_{\gamma,\sigma} &= -\frac{L}{V}\sin\sigma
\end{aligned} \tag{9.16}$$

These terms depend completely on the states and their derivatives. It is now possible to use this information to synthesize the SMC controller.

9.2.2 Definition of Sliding Surfaces

As we previously stated, the objective is to track the given altitude and velocity. In a similar fashion to what has been done in other applications of SMC[75], let us define the decoupled sliding surfaces s_h and s_v,

$$\begin{aligned}
s_h &= \sum_{k=0}^{3} \lambda_k^h \frac{d^{k-1}}{dt^{k-1}} e_h \\
s_V &= \sum_{k=0}^{2} \lambda_k^V \frac{d^{k-1}}{dt^{k-1}} e_V
\end{aligned} \tag{9.17}$$

where the constants λ_i^h and λ_i^V determine the behavior of the closed-loop system, as they define the placement of the eigenvalues of the system during the sliding motion.

The terms e_h and e_V are defined as

$$\begin{aligned} e_h &= h(t) - h_{ref}(t) \\ e_V &= V(t) - V_{ref}(t) \end{aligned} \tag{9.18}$$

It is worth to say that the terms with $k = 0$ in Eq. (9.17) introduce integral terms in the sliding mode equations to improve the performances of the controller. However this is not the only choice, as in other applications[69] these terms are not used and the sums reported in (9.17) begin with $k = 1$. If properly tuned, these elements reduce the overshoot of the system in case of strong initial errors. These relationships define the sliding motion we want to realize. In other words, the induced sliding motion will be described by the homogeneous form of Eq. (9.17). If we develop these equations, we get

$$\begin{aligned} s_h &= \lambda_3^h \ddot{e}_h + \lambda_2^h \dot{e}_h + \lambda_1^h e_h + \lambda_0^h \int_0^t e_h d\tau = 0 \\ s_V &= \lambda_2^V \dot{e}_V + \lambda_1^V e_V + \lambda_0^V \int_0^t e_V d\tau = 0 \end{aligned} \tag{9.19}$$

Equations (9.19) are initially not satisfied. We need to find two candidate Lyapunov functions V_h and V_V; an excellent way to find good candidate functions is by using quadratic functions of the sliding surfaces s_h and s_V.

$$V_m = \tfrac{1}{2} s_m^2, \quad m = h, V \tag{9.20}$$

This choice is motivated by the properties a Laypunov function has to have: if we can demonstrate that for these two functions, we have

$$\begin{aligned} V_m(t) &\geq 0, && \forall t \geq 0 \\ V_m(t) &\to \infty, && m \to \infty \\ \frac{dV_m(t)}{dt} &\leq 0, && \forall t \geq 0, \\ & m = h, V \end{aligned} \tag{9.21}$$

therefore, V_h and V_V are Lyapunov functions, and the nonlinear system described by Eq. (9.15) is globally stable, and for the quadratic functions in many cases it is easy to satisfy these conditions. Indeed, $V_m \to 0$ implies that $s_m \to 0$ and $\dot{s}_m \to 0$, therefore the generic signal $m(t) \to m_{ref}(t)$.

For our functions, more specifically, we are interested to modify the third relationship in Eq. (9.21) as

$$\tfrac{dV_m(t)}{dt} = s_m \dot{s}_m = -k_m \left| s_m \right|, \quad m = h, V \tag{9.22}$$

with k_h, k_v which must be positive. We are assigning to the Lyapunov functions a specific rate of convergence to 0. These conditions, called attractiveness equations, ensure that the distance between the trajectory and the sliding surfaces continuously decreases along the trajectory itself. The practical consequence is that the trajectory

will reach the surfaces in a finite time t_r. After this time, the trajectory "slides" around the surfaces, and the motion described by (9.19) is realized. The times t_r necessary to reach the surfaces depend on the magnitude of the gains k_h and k_v. Equations (9.22) can be replaced by

$$\begin{aligned} \dot{s}_h &= -k_h \operatorname{sign}(s_h) \\ \dot{s}_V &= -k_V \operatorname{sign}(s_v) \end{aligned} \tag{9.23}$$

From the implementation of these equations it will be possible to extract the nonlinear signals which represent our Sliding Mode Controller. These equations are developed in the next section.

9.2.3 Sliding Mode Control Design

Let us replace the left terms of Eq. (9.23) with the relationships developed in Eq. (9.17). We obtain the following equations,

$$\begin{aligned} \lambda_3^h \dddot{e}_h + \lambda_2^h \ddot{e}_h + \lambda_1^h \dot{e}_h + \lambda_0^h e_h &= -k_h \operatorname{sign}(s_h) \\ \lambda_2^v \ddot{e}_V + \lambda_1^V \dot{e}_V + \lambda_0^V e_V &= -k_V \operatorname{sign}(s_V) \end{aligned} \tag{9.24}$$

which link the dynamics of the output error e_h and e_V to the function sign defined as

$$\operatorname{sign}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}, \tag{9.25}$$

This is the keypoint of the method. Once the sliding surface is reached, the trajectory would naturally tend to move away. The high-frequency activity induced by the nonlinear terms in Eq. (9.24) would suddenly react with a command, which has opposite sign w.r.t. the error on the tracked variable. The trajectory would theoretically oscillate with infinite frequency and infinitesimal amplitude around the sliding surfaces. Since every system runs with a given finite frequency, the result is a zig-zag motion with amplitude dependent on the running frequency of the system. This phenomenon takes the name of chattering[54,72], and is, of course, undesirable. However, there are different ways to limit it or even completely remove it.

The most common way, which works very well in practice, is to define a thin layer having width δ_w so that Eqs. (9.24) and (9.25) are modified as follows

$$\lambda_3^h \dddot{e}_h + \lambda_2^h \ddot{e}_h + \lambda_1^h \dot{e}_h + \lambda_0^h e_h = -k_h \operatorname{sat}(s_h) \lambda_2^V \ddot{e}_V + \lambda_1^V \dot{e}_V + + \lambda_0^V e_V \quad = - k_V \operatorname{sat}(s_V) \tag{9.26}$$

with

$$\operatorname{sat}(x) = \begin{cases} 1, & x \geq \delta_w \\ -1, & x \leq -\delta_w \\ \frac{x}{\delta_w}, & -\delta_w < x < \delta_w \end{cases} \tag{9.27}$$

If we combine Eq. (9.15) with Eq. (9.26), and, assigning $\lambda_3^h = 1$, $\lambda_2^V = 1$, we obtain

$$\begin{aligned} a_h + b_{h,\alpha} u_\alpha + b_{h,\sigma} u_\sigma - \dddot{h}_{ref} + \lambda_2^h \ddot{e}_h + \lambda_1^h \dot{e}_h + \lambda_0^h e_h &= -k_h \operatorname{sat}(s_h) \\ a_V + b_{V,\alpha} u_\alpha + b_{V,\sigma} u_\sigma - \ddot{V}_{ref} + \lambda_1^V \dot{e}_V + \lambda_0^v e_V &= -k_V \operatorname{sat}(s_V) \end{aligned} \tag{9.28}$$

To express these equations in matrix form, let us define the following matrices and vectors:

$$\mathbf{u} = \begin{bmatrix} u_\alpha \\ u_\sigma \end{bmatrix} \tag{9.29}$$

$$\mathbf{A} = \begin{bmatrix} a_h \\ a_V \end{bmatrix} \tag{9.30}$$

$$\mathbf{B} = \begin{bmatrix} b_{h,\alpha} & b_{h,\sigma} \\ b_{V,\alpha} & b_{V,\sigma} \end{bmatrix} \tag{9.31}$$

$$\mathbf{\Lambda} = \begin{bmatrix} \Lambda_h \\ \Lambda_V \end{bmatrix} = \begin{bmatrix} -\dddot{h}_{ref} + \lambda_2^h \ddot{e}_h + \lambda_1^h \dot{e}_h + \lambda_0^h e_h \\ -\ddot{V}_{ref} + \lambda_1^V \dot{e}_V + \lambda_0^V e_V \end{bmatrix} \tag{9.32}$$

$$\mathbf{K} = \begin{bmatrix} K_h \\ K_V \end{bmatrix} = \begin{bmatrix} k_h \,\mathrm{sat}(s_h) \\ k_V \,\mathrm{sat}(s_V) \end{bmatrix} \tag{9.33}$$

With these definitions, Eq. (9.28) becomes

$$\mathbf{A} + \mathbf{B}\mathbf{u} + \mathbf{\Lambda} = -\mathbf{K} \tag{9.34}$$

Therefore, we can compute the feedback control rates $\mathbf{u}$ as

$$\mathbf{u} = \mathbf{B}^{-1}\left(-\mathbf{A} - \mathbf{\Lambda} - \mathbf{K}\right) \tag{9.35}$$

and these contributions can be integrated, and added to the reference controls $\alpha_{ref}(t)$ and $\sigma_{ref}(t)$. Therefore, it is possible to obtain the angle of attack and the bank angle as

$$\begin{aligned} \alpha(t) &= \alpha_{ref}(t) + \Delta\alpha(t) = \alpha_{ref}(t) + \int_{t_0}^{t} u_\alpha d\tau \\ \sigma(t) &= \sigma_{ref}(t) + \Delta\sigma(t) = \sigma_{ref}(t) + \int_{t_0}^{t} u_\sigma d\tau \end{aligned} \tag{9.36}$$

9.2.4 Applicability, Stability, Implementation Aspects and Reduction to 1-DOF Controls

From the inspection of Eq. (9.35) it is possible to see that the control can be synthesized only if the matrix $\mathbf{B}$ is nonsingular. This happens if the determinant of $\mathbf{B}$ is different than 0.

$$\det(\mathbf{B}) = b_{h,\alpha} b_{V,\sigma} - b_{h,\sigma} b_{V,\alpha} \neq 0 \tag{9.37}$$

If we look at the definitions given in Eq. (9.16), we can write

$$\det(\mathbf{B}) = 0 \Leftrightarrow b_{D,\alpha} L \cos\gamma \sin\sigma = 0 \tag{9.38}$$

Since $b_{D,\alpha}$ and L are always different from 0, from a physical point of view Eq. (9.38) gives us the two only possibilities where the control synthesis cannot be applied. The

former is related to the condition $\gamma = \pm 90$ deg, which means that these equations are not valid for vertical motion. This condition is excluded during the trajectory planning in any case. The latter is related to the condition $\sigma = 0$ deg, which represents a well-known singularity in the entry dynamics[78,79,80]. To exclude this possibility, a region around $\sigma = 0$ deg is excluded, except during the bank reversals. This choice has also another reason associated with the lateral performance of the system. A trade-off between longitudinal and lateral performance is needed. Without lateral control, the longitudinal errors are reduced, but this may cause larger lateral dispersions, as the heading error has no effect on the bank-angle determination. A too aggressive lateral controller may cause frequent bank reversals, which may yield a deterioration of the longitudinal performance. For this specific mission, since we work with a strongly asymmetric scenario, a trade-off between lateral and longitudinal performances has suggested to exclude the interval $[-12.5, 1]$ deg.

The feedback controller is triggered at the end of the reference modulation of the angle of attack. The proposed longitudinal tracking controller allows to easily limit the control rates together with the controls. Moreover, it is possible to define a relative saturation on the angle of attack to limit the variation to a given range (for the case proposed here $\Delta\alpha \in [-2.5, 10]$ deg). With these definitions, it is possible to track the altitude and the velocity by modulating the bank angle σ and the angle of attack α at the same time. A significant advantage of this formulation is its reduced size. It is based on a small number of constant parameters (λ_i^j, k_i, and δ_w^i), therefore no gain-scheduling techniques need to be applied, with a consistent save in terms of memory required on-board. Moreover, despite what happens with H_∞ controllers and other linear techniques, the stability of the nonlinear system is ensured directly by the convergence of the Lyapunov functions.

An example of the Lyapunov functions V_h and V_v is depicted in Figs. 9.1 and 9.2. In these examples the feedback control law of the vehicle begins after 88 s. During the first 88 seconds, the Lyapunov function follows an oscillating behavior. When the controller is triggered, the functions V_h and V_V tend to monotonically decrease, except during the bank reversal or in case the controls saturate. In this case, a transient with small increases of the functions is observed, before they begin to decrease again.

Moreover, the 2-DOF controller can be reduced to several 1-DOF controllers by selecting the variable to track and the control to use. For instance, only the bank angle could be used to track the altitude for missions, where modulation of the angle of attack is not possible. Table 9.1 shows the possible combinations, which can be obtained from the proposed general controller. The first column determines the variable to track, the second and third columns show which specific control inputs can be used (1) or not (0).

Finally, the fourth column explains the guidance mode associated with that specific row. For instance, if we look at the fourth row we can see that the altitude can be tracked by only using the bank angle. Then, only the related rows and columns of the Eqs. (9.29)-(9.36) must be selected. These properties make the use of SMC an interesting technique for this class of problems.

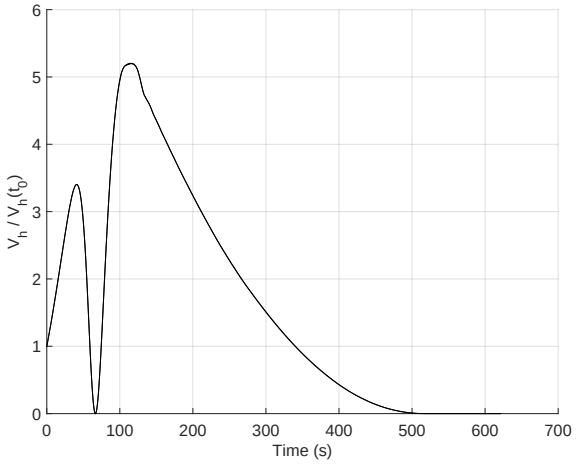


Figure 9.1: Example of convergence of Lyapunov function V_h - controlled motion begins at 88 s.

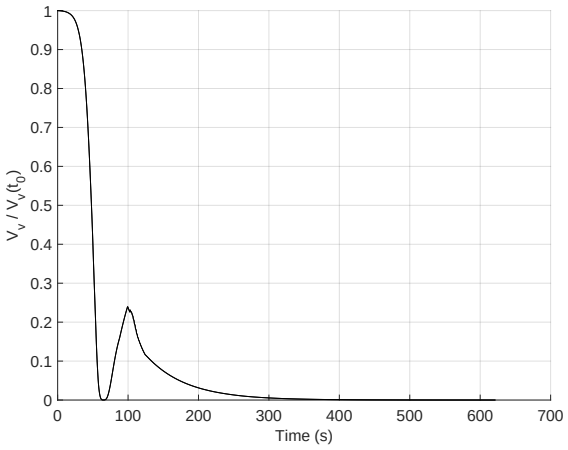


Figure 9.2: Example of convergence of Lyapunov function V_V - controlled motion begins at 88 s.

9.3 Lateral Controller

The lateral control is usually based on the heading error, defined as the difference between the velocity-azimuth angle ψ (positive when flying towards local north), and the plane containing the origin of the reference frame, the current position and the target position of the vehicle. If the heading error is equal to 0 deg, the vehicle is flying exactly towards the target. However, it may happen that during the flight one or more lateral corrections become necessary. The first step is to compute, during the trajectory, the heading error; this can be done through several algorithms[2,37,81].

Table 9.1: Guidance-mode matrix.

Variable(s) to track	α	σ	Mode
h, V	0	0	MIMO open-loop
h, V	1	1	MIMO closed-loop
h	1	0	Altitude tracking (e.g., bank-reversal, bank-angle saturation)
h	0	1	Altitude tracking (e.g. fixed α, α saturation)
V	1	0	Velocity tracking (not used here)
V	0	1	Not physically possible ($b_{V,\sigma} = 0$)

Alternatively, the haversine formula can be used. This relationship relates the arc d on a sphere of radius R formed by two points, having longitudes θ_1 and θ_2 and latitudes ϕ_1 and ϕ_2, as shown in Fig. 9.3.

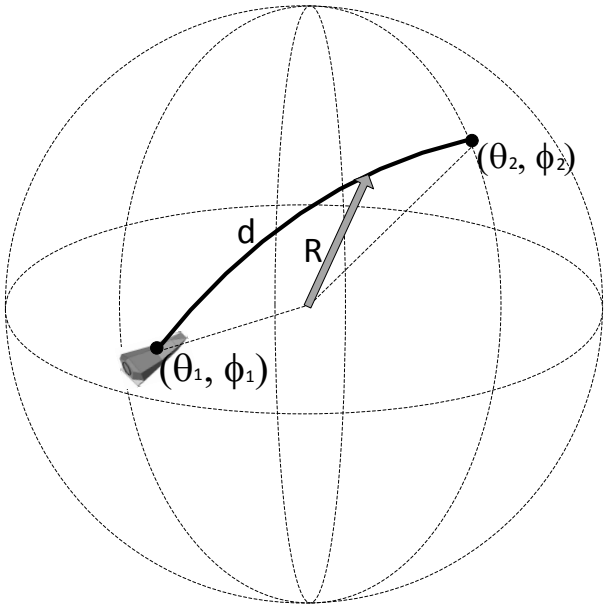


Figure 9.3: Arc computed through haversine formula.

$$\text{haversin}\left(\frac{d\,(\theta_1,\phi_1,\theta_2,\phi_2)}{R}\right) = \text{haversin}(\phi_2-\phi_1)+\cos\phi_1\cos\phi_2\,\text{haversin}(\theta_2-\theta_1) \quad (9.39)$$

The function haversin is defined as

$$\text{haversin}\,\theta = 1-\frac{\cos\theta}{2} = \sin^2\frac{\theta}{2} \quad (9.40)$$

It is possible, for each couple of points (θ_1, ϕ_1) and (θ_2, ϕ_2), to compute the length of the arc d as

$$d\left(\theta_1, \phi_1, \theta_2, \phi_2\right) = 2R\sin^{-1}\left[\left(1 - \frac{\cos\left(\phi_2 - \phi_1\right)}{2}\right) + \cos\phi_1 \cos\phi_2 \left(1 - \frac{\cos\left(\theta_2 - \theta_1\right)}{2}\right)\right]^{1/2} \tag{9.41}$$

Let us define a spherical triangle having the following sides.

$$\begin{array}{rcl} \widehat{A} & = & \left(\phi_f - \phi\right) \\ \widehat{B} & = & \left(\theta_f - \theta\right) \\ \widehat{C} & = & d\left(\theta, \phi, \theta_f, \phi_f\right) \end{array} \tag{9.42}$$

By using the law of sines, it is possible to obtain the angle between the direction of motion and the side $\widehat{B}$ as

$$\psi_{HOR} = \begin{cases} \sin^{-1}\frac{\sin\widehat{A}}{\sin\widehat{C}} & \widehat{B} \geq 0 \\ \pi - \sin^{-1}\frac{\sin\widehat{A}}{\sin\widehat{C}} & \widehat{B} < 0 \end{cases} \tag{9.43}$$

Since ψ is defined w.r.t the local north direction, the line of sight (LOS) of the vehicle w.r.t. the target point is defined as

$$\psi_{LOS} = \frac{\pi}{2} - \psi_{HOR} \tag{9.44}$$

We can compute the heading error as

$$\Delta\psi = \psi - \psi_{LOS} \tag{9.45}$$

This formulation is mathematically equivalent to the vector-based one, but computationally lighter. A series of tests has shown that the proposed formulation can be between 3 and 300 times faster than the classical formulation, with no loss of information.

A graphical representation of the heading error is reported in Fig. 9.4. It is possible to see that the angle ψ_{LOS} is the angle between the local north direction and the current direction of flight represented by the side $\widehat{C}$ of the spherical triangle $\widehat{A}\widehat{B}\widehat{C}$.

The heading error can be then kept under control by applying a bank reversal when it exceeds a given threshold.

$$|\Delta\psi| \geq \Delta\psi_{limit} \tag{9.46}$$

When a bank reversal is commanded, the bank-angle rate is set equal to the maximum bank-angle rate, multiplied by the opposite sign of the current bank angle. The angle of attack rate is derived from the third row of the guidance-mode matrix reported in Table 9.1, which leads to the following equation,

$$\begin{bmatrix} u_\alpha \\ u_\sigma \end{bmatrix} = \begin{bmatrix} b_{h,\alpha}^{-1}\left(-a_h - \Lambda_h - K_h\right) \\ -\text{sign}(\sigma_{cur})u_{\sigma,max} \end{bmatrix} \tag{9.47}$$

where σ_{cur} is the current value of the bank angle, that is, the bank angle at the moment of the beginning of the maneuver.

Initially the heading error is larger than the corridor, and decreases over time. Therefore, the lateral guidance can be applied only when the heading error changes sign at least once. An example of corridor and bank reversal is shown in Fig. 9.5. A first reversal is commanded if the heading error becomes larger than 12 deg. After the first reversal, the heading error corridor is enlarged to ±20 deg, in a similar fashion to what done for the Space Shuttle[2]. The specific values have been selected after several tests, and represent a satisfying trade-off between lateral performance (where theoretically more bank reversals improve the performance), and longitudinal performance (which suggest to reduce the number of bank reversals as much as possible), and in general may differ according to the mission profile, and the specific vehicle.

If a second bank reversal is commanded, the heading corridor is enlarged to ±180 deg, to prevent other bank reversals. In other words, the maximum number of allowed bank reversals is equal to 2. With reference to Fig. 9.5, it is possible to observe that in some cases the heading error switches suddenly from almost 0 deg to almost 180 deg. The reason for this behavior is that, for cases where the actual downrange is larger than the nominal one, the velocity has opposite direction w.r.t. the vector connecting the current vehicle position and the terminal position. For this reason the heading angle becomes larger than 90 deg, and tends to 180 deg when the crossrange error tends to 0. This is only a mathematical implication of using the heading error as reference parameter for the lateral guidance, and has no practical consequences on the results.

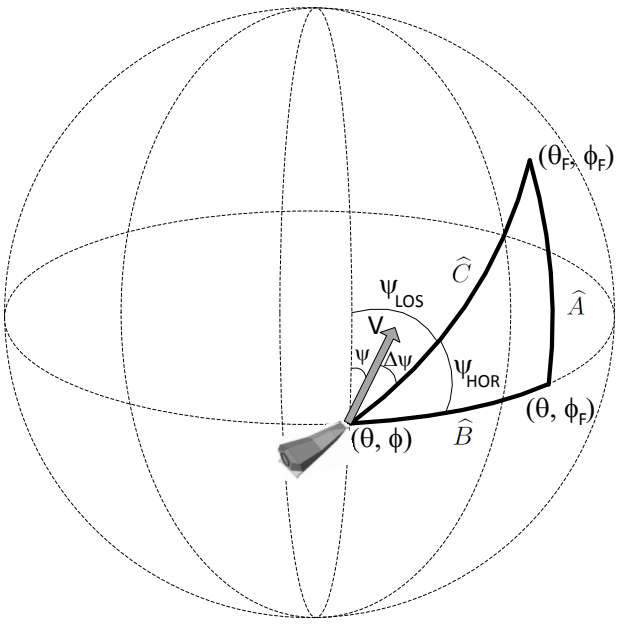


Figure 9.4: Heading error computed through spherical trigonometry.

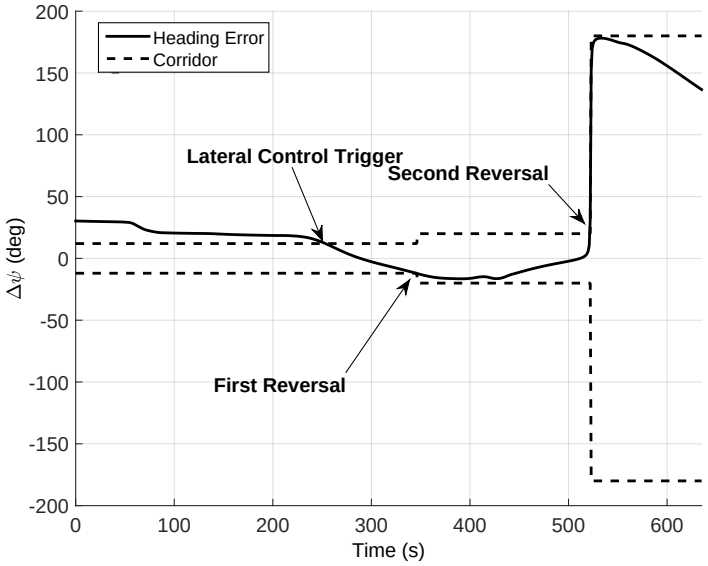


Figure 9.5: Heading error corridor.

9.4 Nominal Solution

In this section a nominal closed-loop simulation is compared with the open-loop behavior of the solution. The initial conditions are taken from Table 2.3. States are plotted in Fig. 9.6, while controls are depicted in Fig. 9.7. The states are perfectly consistent, having an error on the states in the order of between 10^{-7} and 10^{-9} for all the states. This demonstrates that the behavior of the equivalent input/output differential equations defined in Eq. (9.28) is fully consistent with the initial differential system described by Eq. (4.12), and since we are dealing with nominal conditions, no big efforts in terms of controls are required. Therefore, the control profiles plotted in Fig. 9.7 associated with the closed-loop system are essentially the same as the open loop, and the same conclusion can clearly be drawn for the constraints (Fig. 9.9), and the groundtrack (Fig. 9.8).

The behavior shown here is in nominal conditions. Off-nominal conditions, together with uncertainties and disturbances will affect the results in more significant way. The feedback controller will mitigate these effects such that all the mission requirements are satisfied.

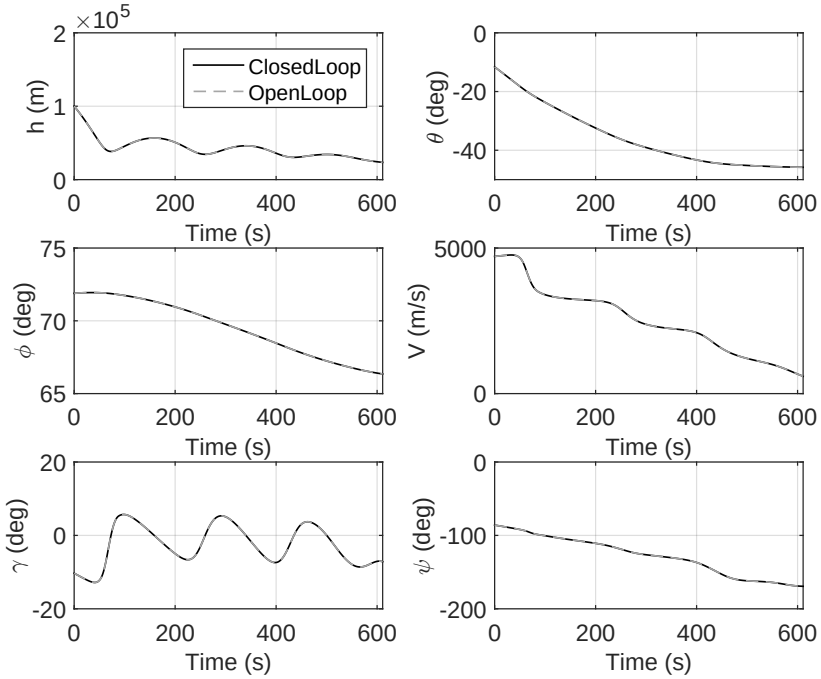


Figure 9.6: Nominal scenario - states.

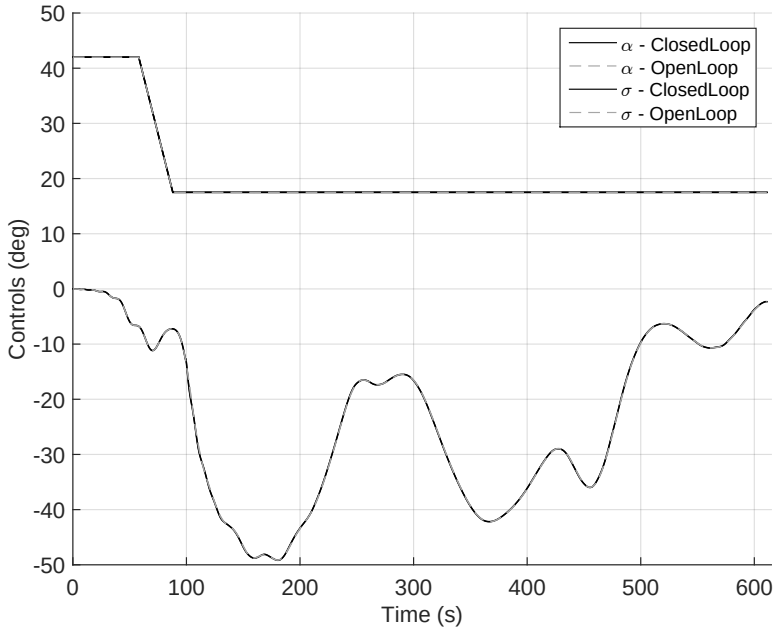


Figure 9.7: Nominal scenario - controls.

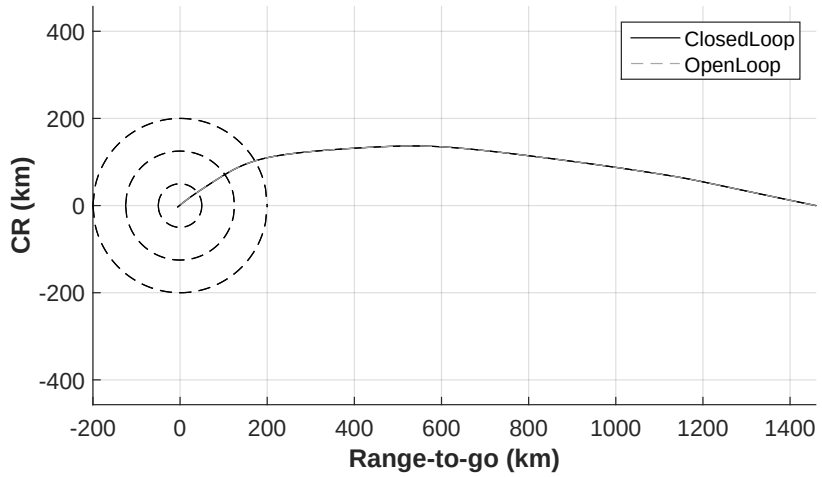


Figure 9.8: Nominal scenario - groundtrack.

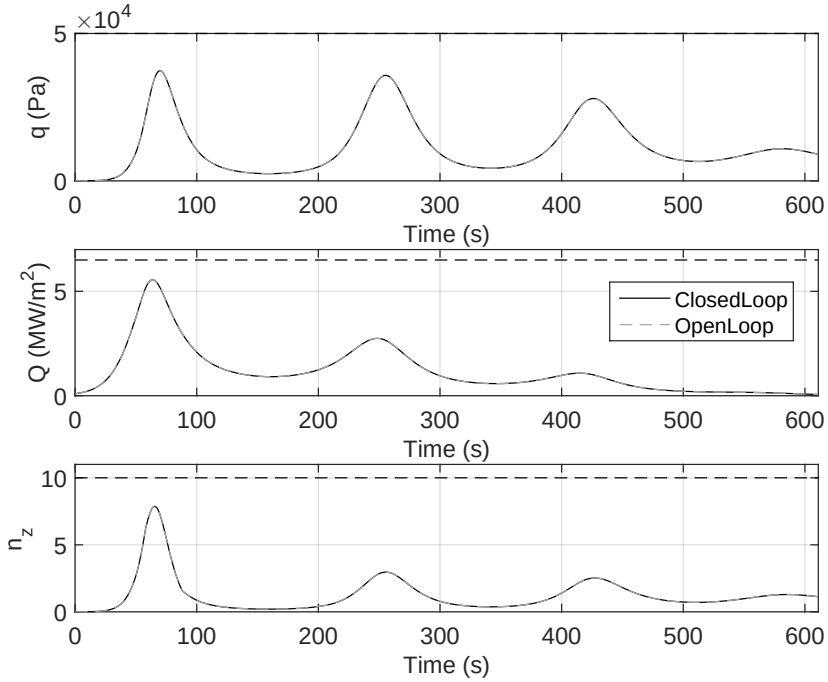


Figure 9.9: Nominal scenario - constraints.

Chapter 10

Backup Guidance - Drag Energy Guidance

This chapter describes the backup guidance scheme implemented for SHEFEX-3. Even though this guidance scheme is not required for the thesis directly, it has been developed for two reasons. The first is associated with the project requirement Req-G036, taken from Table 2.4, where it is stated that the guidance system needs to be redundant. The second reason is that the guidance scheme proposed in this section is based on the traditional methods implemented over the years for several missions [1,2,7,37,81], and therefore represents a good benchmark to quantify the results obtained with the main guidance proposed in Chapters 7-9, and compared with a more traditional approach (this comparison will be done in Sec. 11.6). Specifically, the backup guidance is derived from the Space Shuttle guidance [2], and its successive evolutions [7,37,82].

In all these cases the central idea is to represent the longitudinal constraints, such as heat-flux, dynamic pressure and load-factor in the drag-energy plane. A drag-energy reference profile (which can be adaptive or not) can be implemented, and from this profile and its derivatives w.r.t. the energy, the reference bank angle, representing the feedforward guidance command, can be extracted. A tracking controller is added to provide stability to the system. However, several differences w.r.t. the models presented in the cited sources are included for SHEFEX-3. For instance, all the cited works rely on several assumptions, which simplify the guidance equations, such as the use of exponential atmosphere. In this case, instead, the use of more realistic models, such as the US76 standard atmosphere, introduced in Sec. 3.1.2, or a more complex aerodynamic database, as treated in Sec. 3.4, makes these simplifications not applicable. Therefore, extra terms need to be taken into account, leading to equations which include the previous ones as special cases.

In particular, to take advantage from the use of SPARTAN, and to keep the design as simple as possible, the feedforward guidance is derived from the nominal solution, while for the feedback controller a drag-dynamics inversion method has been used. It is worth to notice that, while for the Space Shuttle and other vehicles it is possible to model the drag-energy profile with simple analytical expressions (e.g. linear, quadratic,

etc., as described in Sec. 5.1.1), for SHEFEX-3 this is not possible. This is due to a series of factors. Given the initial flight-path angle, and the limits of the controls, together with the need to cover the prescribed range, an oscillated entry has to be followed, as one can see in Fig. 8.12. This implies that the flight-path angle follows an oscillating behavior (the flight-path angle directly affects the range, in virtue of the differential equation $\dot{R} = V \cos\gamma$), and so does the altitude. The optimal drag acceleration profile follows an oscillating behavior too, and this implies that the signs of the derivatives of C_D change more than once. This makes the use of simple analytical profiles to describe it extremely hard. For this reason as well, the logic behind the Space Shuttle guidance has to be reviewed, to take the SHEFEX-3 scenario properly into account. Consequently, the proposed formulation of guidance system is generally applicable to other vehicles, missions, and even celestial bodies (e.g. Mars entry missions).

The scheme adopted is depicted in Fig. 10.1. The chapter is organized as follows: in Sec. 10.1 the determination of the drag-energy feasible space is described. In Sec. 10.2 the equations of motion are transformed into the corresponding drag-energy dynamics. The results are used to design the feedforward (Sec. 10.3) and the feedback (Sec. 10.4) guidance scheme implemented as backup guidance. The use of the angle of attack as aid is described in Sec. 10.5, while the nominal behavior of the system in closed and open loop is described and shown in Sec. 10.6.

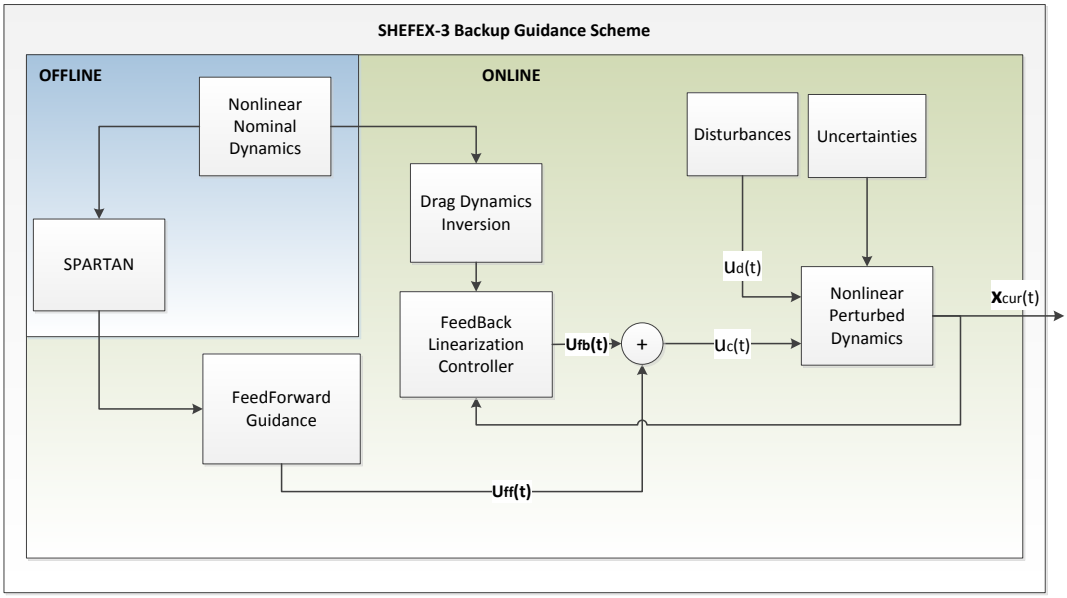


Figure 10.1: Backup guidance scheme.

In Fig. 10.1 we can distinguish two subsystems. The first system is the feedforward guidance, which provides reference commands to the system. As will be explained in this chapter, this module relies on the optimal solution computed in nominal conditions with

SPARTAN. The full nonlinear system is taken into account, and used by SPARTAN to compute the optimal reference drag profile, and the reference bank-angle commands. The second subsystem is represented by the feedback controller. The nonlinear dynamics is transformed into a feedback-linearized system by using the drag dynamics, and the current measurements of drag to improve the performance of the system.

10.1 Drag-Energy feasibility space

The first operation to perform is the determination of the feasible drag-energy space. As feasible space in this context we mean those points on the drag-energy plane, which do not cause any constraint violation, (e.g., a too-high load factor or heat-flux). Indeed, one can observe that, in the hypothesis of nominal angle of attack, $\alpha = \alpha(E)$, all the constraints are a function of altitude and velocity only. E is the specific mechanical energy of the system, equal to $\frac{V^2}{2} - \frac{\mu_\oplus}{r}$, and is a function of altitude and velocity too, while $\mu_\oplus$ is the Earth gravitational parameter. Therefore, for each of the constraints, and each energy level, we can solve a system of two nonlinear equations, obtaining the altitude and velocity, which generate the maximum value of a specific constraint. Since α is a given function of energy, we can compute the corresponding drag acceleration. If we exceed that specific value at that energy level, the specific constraint will be violated. This process can be done for each of the three constraints considered here, and is performed by using the bisection method to solve the systems of nonlinear equations.

10.1.1 Dynamic pressure

In this case, for each value of the energy E, the two equations to be solved are

$$\begin{aligned} \frac{V^2}{2} - \frac{\mu_\oplus}{h + r_\oplus} &= E \\ \frac{1}{2}\rho(h)V^2 &= \bar{q}_U \end{aligned} \tag{10.1}$$

where q_U is the maximum value allowed for the dynamic pressure. The solution to Eq. (10.1) will give us the altitude h_q and velocity V_q profiles, which maximize the dynamic pressure. The corresponding drag acceleration is

$$D_{\bar{q},max} = \frac{1}{2}\rho(h_q){V_q}^2\frac{S}{m}C_D(\alpha, h_q, V_q) \tag{10.2}$$

10.1.2 Heat flux

In this case the two equations to be solved are

$$\begin{aligned} \frac{V^2}{2} - \frac{\mu_\oplus}{h + r_\oplus} &= E \\ k_Q\sqrt{\rho(h)}V^3 &= \dot{Q}_U \end{aligned} \tag{10.3}$$

where $\dot{Q}_U$ is the maximum value allowed for the heat-flux. The coefficient k_Q is a constant depending on the material and the geometry of the thermal protection system, for SHEFEX-3 equal to 1.2444 10^{-3} $\mathrm{kg}^{1/2}/\mathrm{m}^3$. The heat-flux is computed by using the cold-wall model for laminar boundary layer). In this case the solution to Eq. (10.3) will give us the altitude h_Q and velocity V_Q profiles which maximize the heat flux. The corresponding drag acceleration is

$$D_{\dot{Q},max} = \frac{1}{2}\rho(h_Q){V_Q}^2\frac{S}{m}C_D(\alpha, h_Q, V_Q) \tag{10.4}$$

10.1.3 Vertical load factor

The third constraint taken into account is the vertical load factor. The corresponding equations to be solved are

$$\begin{aligned} \frac{V^2}{2} - \frac{\mu_\oplus}{h + r_\oplus} &= E \\ \frac{|L(\alpha, h, V)\cos\alpha + D(\alpha, h, V)\sin\alpha|}{g_0} &= n_{z,U} \end{aligned} \tag{10.5}$$

where $n_{z,U}$ is the maximum value allowed for the vertical load factor, L and D are the lift and drag acceleration, respectively, and g_0 is the gravity acceleration at sea level. In this case the solution to Eq. (10.5) will give us the altitude h_{n_z} and velocity V_{n_z} profiles, which maximize the vertical load factor. The corresponding drag acceleration is

$$D_{n_z,max} = \frac{1}{2}\rho(h_{n_z}){V_{n_z}}^2\frac{S}{m}C_D(\alpha, h_{n_z}, V_{n_z}) \tag{10.6}$$

Notice that no restrictive hypothesis on the atmospheric model has been formulated so far. Therefore, Eqs. (10.1), (10.3), and (10.5) are completely general. Once the analysis is completed, the drag-energy feasible space can be defined as

$$D, E : D_{Eq_g} \leq D(E) \leq \min \begin{cases} D_{\bar{q},max} \\ D_{\dot{Q},max} \\ D_{n_z,max} \end{cases} \tag{10.7}$$

Figure 10.2 shows the feasible space derived for SHEFEX-3, obtained as intersection of the several constraints, represented as drag-energy boundaries. In this plot the drag acceleration has been normalized w.r.t. the gravity acceleration at sea-level, while as already done in other works[7,37,38] the energy has been scaled according to

$$e = \frac{E - E_0}{E_F - E_0} \tag{10.8}$$

so that it is equal to 0 at the beginning of the entry, and equal to 1 at the TAEM. From the analysis of Fig. 10.2 we can observe that in the early phase of the atmospheric entry, the heat-flux is the most challenging constraint. During the modulation of the angle of attack (which goes from e $\cong$ 0.1 to about e $\cong$ 0.5), the load factor becomes the dominant

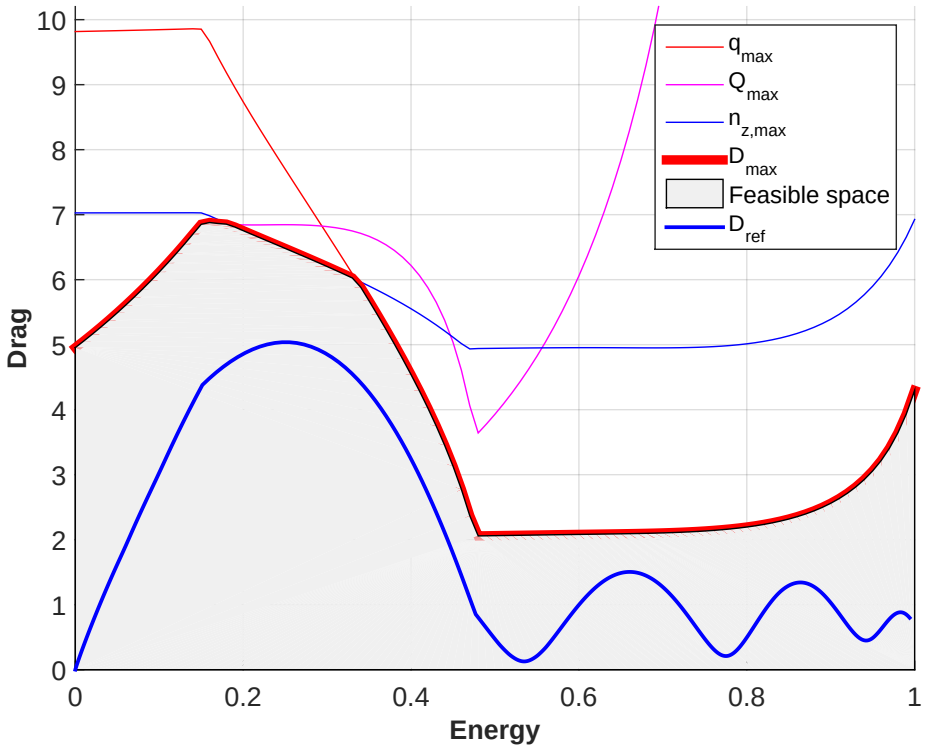


Figure 10.2: Drag-energy feasible space.

factor, and from that point until the end of the mission, the higher values of the density make the dynamic pressure the dominant constraint. The blue line represents the optimal solution found with SPARTAN, which satisfies all the constraints.

It is now possible to derive the drag dynamics equations for the development of the backup guidance scheme.

10.2 Drag-Energy Dynamics

Let us consider the longitudinal equations of motion given by Eq. (4.16).

$$\begin{aligned} h' &= -\frac{\sin\gamma}{D} \\ V' &= \frac{1}{V} + \frac{g}{DV}\sin\gamma \\ \gamma' &= \frac{\cos\gamma}{V^2}\left(g - \frac{V^2}{r}\right)\frac{1}{D} - \frac{1}{V^2}u \end{aligned} \tag{10.9}$$

where the control u is defined as

$$u = \frac{L}{D}\cos\sigma \tag{10.10}$$

The independent variable is the specific energy, $E = \frac{V^2}{2} - \left(\frac{\mu_\oplus}{r}\right)$, and the energy dissipation can be expressed as

$$\dot{E} = -DV \tag{10.11}$$

We are interested to develop a guidance system able to track the drag in the energy domain. In a similar fashion to what has been done in literature[2,7,37], we can use a feedback-linearization technique, which allows us to control the second derivative of D with the control u. The drag-dynamics will be expressed as a second-order system. Let us now differentiate the drag acceleration w.r.t. energy.

$$\begin{aligned} D &= \frac{1}{2}\rho V^2 \frac{S}{m} C_D \\ D' &= D\left(\frac{\rho'}{\rho} + 2\frac{V'}{V} + \frac{{C_D}'}{C_D}\right) \end{aligned} \tag{10.12}$$

For the second derivative, we get

$$D'' = D'\left(\frac{\rho'}{\rho} + 2\frac{V'}{V} + \frac{{C_D}'}{C_D}\right) + D\left(\frac{\rho''}{\rho} - \frac{\rho'^2}{\rho^2} + 2\frac{V''}{V} - 2\frac{V'^2}{V^2} + \frac{C_D''}{C_D} - \frac{{C_D'}^2}{C_D^2}\right) \tag{10.13}$$

From the analysis of Eq. (10.13), we can see that we need some other derivatives as well. In the next sections, each of the contributions will be derived. Specifically, these contributions involve the atmospheric density, the aerodynamic coefficients, the Mach number, and the angle of attack.

10.2.0.1 Atmospheric density

Equation (10.13) requires the computation of the derivative of the atmospheric density. In the case of the Space Shuttle entry guidance[2], or in several drag-energy methods[7,37,52], since the atmospheric density is modeled as exponential profile $\rho_0 e^{-\frac{h}{h_s}}$, with h_s scale height, its derivative w.r.t. energy is

$$\rho' = -\rho\frac{h'}{h_s} \tag{10.14}$$

and since $\frac{\rho'}{\rho}$ is simply $-\frac{h'}{h_s}$, if we derive the two terms w.r.t. energy, we simply get

$$\frac{\rho''}{\rho} - \frac{\rho'^2}{\rho^2} = -\frac{h''}{h_s} \tag{10.15}$$

In the most general case we have

$$\begin{aligned} \rho' &= \rho_h h' \\ \rho'' &= \rho_{hh} h'^2 + \rho_h h'' \end{aligned} \tag{10.16}$$

where ρ_h and ρ_{hh} are the first and second derivatives of the atmospheric density w.r.t. the altitude, which can come from either an analytical or a numerical model. Therefore, we can write

$$\frac{\rho''}{\rho} - \frac{\rho'^2}{\rho^2} = \frac{\rho_{hh} h'^2 + \rho_h h''}{\rho} - \frac{\rho_h^2 h'^2}{\rho^2} \tag{10.17}$$

Clearly Eq. (10.17) can be applied to any atmospheric model, and reduces to Eq. (10.15) when the hypothesis of exponential atmospheric density is assumed.

10.2.0.2 Aerodynamics

Equation (10.13) includes the derivative of the drag coefficient as well. For some vehicles a common assumption usually made is the following:

$$\begin{aligned} C_D' &= \bar{C}_{D,i}' \; i = 1, \cdots, N \\ C_D'' &= 0 \end{aligned} \tag{10.18}$$

where $\bar{C}_{D,i}'$ is a constant, or a series of constants[7] according to the number of values N chosen to model it, or, in some cases, can be assumed equal to 0[38,42]. In general however, since the drag coefficient C_D depends on the angle of attack α, the Mach number M, and the altitude h, its derivatives are

$$\begin{aligned} C_D{}' &= C_{D,\alpha}\alpha' + C_{D,M}M' + C_{D,h}h' \\ C_D{}'' &= C_{D,\alpha\alpha}\alpha'^2 + C_{D,\alpha M}\alpha' M' + C_{D,\alpha h}\alpha' h' + C_{D,\alpha}\alpha'' + \\ &+ C_{D,M\alpha}M'\alpha' + C_{D,MM}M'^2 + C_{D,Mh}M'h' + C_{D,M}M'' + \\ &+ C_{D,h\alpha}h'\alpha' + C_{D,hM}h'M' + C_{D,hh}h'^2 + C_{D,h}h'' \end{aligned} \tag{10.19}$$

The derivatives $C_{D,i}$ and $C_{D,jk}$ are the first and second derivatives of the drag coefficient w.r.t. the variables i, and j, k, respectively. These terms are computed with central differences. In this case too, they reduce to Eq. (10.18) if the same assumptions are made.

10.2.0.3 Mach number

For the Mach number derivatives we can write

$$\begin{aligned} M' &= M_h h' + M_V V' \\ M'' &= M_{hh}h'^2 + 2M_{Vh}h'V' + M_h h'' + M_{VV}V'^2 + M_V V'' \end{aligned} \tag{10.20}$$

The derivatives of the Mach number w.r.t. h and V can be obtained analytically from the definition of Mach number, $M = V/\sqrt{\gamma_{air} R^* T}$

$$\begin{aligned} M_h &= -\frac{M}{2}\frac{T_h}{T} \\ M_V &= \frac{1}{\sqrt{\gamma_{air} R^* T}} \end{aligned} \tag{10.21}$$

T_h is the temperature derivative w.r.t. the altitude, and can be obtained numerically from the model used. γ_{air} is the specific heat ratio of the air, assumed equal to 1.4, and R^* is the specific gas constant for Earth, assumed to be equal to 287.058 J/kg K.

10.2.1 Angle of attack

The derivatives of the angle of attack α' and α'', are known, as the function $\alpha(E)$ is given in numerical form.

10.2.2 Feedback linearized dynamics

With the previous contributions defined, it is now possible to define the feedback-linearized dynamics. Let us define the second derivatives of altitude h and velocity V w.r.t. energy. Starting from Eq. (10.9), and differentiating twice we get the following expressions.

$$\begin{aligned} h'' &= \frac{\sin\gamma}{D^2}D' - \frac{\cos\gamma}{D}\gamma' \\ V'' &= -\frac{V'}{V^2} + \frac{g'\sin\gamma}{DV} + \frac{g\cos\gamma\gamma'}{DV} - \frac{g\sin\gamma}{D^2V^2}\left(D'V + DV'\right) \end{aligned} \tag{10.22}$$

Since the required derivatives have been defined, we can build the relationship between the second derivative of drag acceleration D'' and the control u. To do so, we will express all variables in affine form, separating the part, which depends on the control, from the part that does not. For the flight-path angle, we have

$$\gamma' = a_\gamma + b_\gamma u \tag{10.23}$$

with

$$\begin{aligned} a_\gamma &= \frac{\cos\gamma}{D}\left(\frac{g}{V^2} - \frac{1}{r}\right) \\ b_\gamma &= -\frac{1}{V^2} \end{aligned} \tag{10.24}$$

The second derivative of altitude becomes

$$h'' = a_h + b_h u \tag{10.25}$$

with

$$\begin{aligned} a_h &= \sin\gamma\frac{D'}{D^2} - \frac{\cos\gamma}{D}a_\gamma \\ b_h &= -\frac{\cos\gamma}{D}b_\gamma \end{aligned} \tag{10.26}$$

The second derivative of atmospheric density becomes

$$\rho'' = a_\rho + b_\rho u \tag{10.27}$$

with

$$\begin{aligned} a_\rho &= \rho_{hh} h'^2 + \rho_h a_h \\ b_\rho &= \rho_h b_h \end{aligned} \tag{10.28}$$

For the velocity we have

$$V'' = a_V + b_V u \tag{10.29}$$

with

$$\begin{aligned} a_V &= -\frac{V'}{V^2} + \frac{g' \sin\gamma}{DV} + \frac{g \cos\gamma a_\gamma}{DV} - \frac{g \sin\gamma}{D^2 V^2} \left(D'V + DV' \right) \\ b_V &= \frac{g \cos\gamma b_\gamma}{DV} \end{aligned} \tag{10.30}$$

The second derivative of Mach number can be expressed as

$$M'' = a_M + b_M u \tag{10.31}$$

with

$$\begin{aligned} a_M &= M_{hh} h'^2 + 2 M_{Vh} h' v' + M_h a_h + M_{VV} V'^2 + M_V a_V \\ b_M &= M_h b_h + M_V b_V \end{aligned} \tag{10.32}$$

The second derivatives of Mach number are

$$\begin{aligned} M_{hh} &= \frac{3}{4} M \left(\frac{T_h}{T} \right)^2 - \frac{M}{2} \frac{T_{hh}}{T} \\ M_{hV} &= M_{Vh} = -\frac{1}{2} \frac{T_h}{T} \\ M_{VV} &= 0 \end{aligned} \tag{10.33}$$

Finally, for the second derivative of C_D, we have

$$C_D'' = a_{C_D} + b_{C_D} u \tag{10.34}$$

with

$$\begin{aligned} a_{C_D} &= C_{D,\alpha\alpha} \alpha'^2 + C_{D,\alpha M} \alpha' M' + C_{D,\alpha h} \alpha' h' + C_{D,\alpha} \alpha'' + \\ &\quad + C_{D,M\alpha} M' \alpha' + C_{D,MM} M'^2 + C_{D,Mh} M' h' + C_{D,M} a_M + \\ &\quad + C_{D,h\alpha} h' \alpha' + C_{D,hM} h' M' + C_{D,hh} h'^2 + C_{D,h} a_h \\ b_{C_D} &= C_{D,M} b_M + C_{D,h} b_h \end{aligned} \tag{10.35}$$

The second derivative of drag acceleration can thus be expressed as

$$D'' = a_D + b_D u \tag{10.36}$$

with

$$\begin{aligned} a_D &= D' \left(\frac{\rho'}{\rho} + 2\frac{V'}{V} + \frac{{C_D}'}{C_D} \right) + D \left(\frac{a_\rho}{\rho} - \frac{\rho'^2}{\rho^2} + 2\frac{a_V}{V} - 2\frac{V'^2}{V^2} + \frac{a_{C_D}}{C_D} - \frac{{C_D}^2}{C_D^2} \right) \\ b_D &= D \left(\frac{b_\rho}{\rho} + 2\frac{b_V}{V} + \frac{b_{C_D}}{C_D} \right) \end{aligned} \tag{10.37}$$

We can now design the guidance scheme, such that the error in terms of drag-energy is minimized.

10.3 Feedforward Guidance

To develop the feedforward guidance scheme, in general[2,7,42,83] an ad-hoc, simple drag-energy profile is designed to satisfy all the requirements, and at the same time, can easily be adapted to the needs. Once that the reference drag profile D_{ref}, is designed, the feedforward guidance can be computed by manipulating Eq. (10.36).

$$u_{FF} = \frac{D''_{ref} - a_D}{b_D} \tag{10.38}$$

The corresponding bank angle can be computed by

$$\sigma = s_{lat} \cos^{-1}\left(\frac{u_{FF}}{C_L/C_D}\right) \tag{10.39}$$

where D''_{ref} is the second derivative of the reference drag profile w.r.t. energy, while the term s_{lat} is the sign of the bank angle, determined according to the lateral controller strategy. However, given the nature of the mission for SHEFEX-3, a different strategy is followed. Specifically, we are interested to use the drag-energy profile computed with SPARTAN and depicted in Fig. 10.2, and to use it as a reference feedforward solution.

The optimal control problem is set according to the transcription provided in Chapter 8 for the generation of the trajectory-database, by using the nominal initial conditions. Figs. 9.6-9.8 show the corresponding solution over the non-dimensional energy domain. Given the reference bank angle σ_{ref} we can write the feedforward solution as

$$u_{FF} \cong u_{ref} = \left(\frac{C_{L,ref}}{C_{D,ref}}\right) \cos\sigma_{ref} \tag{10.40}$$

where $C_{L,ref}$, $C_{D,ref}$ and σ_{ref} are the lift and drag coefficients, and the bank angle corresponding to the nominal solution. This law will be extended with a feedback term to take off-nominal conditions into account.

10.4 Feedback Guidance

In a similar fashion to what has been done in literature[81,84], let us define the drag error and its derivatives:

$$\begin{aligned} \Delta D &= D - D_{ref} \\ \Delta D' &= D' - D'_{ref} \\ \Delta D'' &= D'' - D''_{ref} \end{aligned} \tag{10.41}$$

The objective is to obtain a combined feedforward-feedback control law:

$$u = u_{FF} + u_{FB} \tag{10.42}$$

where the first term of the right side of Eq. (10.42) is given by Eq. (10.38). We need to determine now u_{FB}. Specifically we are interested to model the error dynamics as a

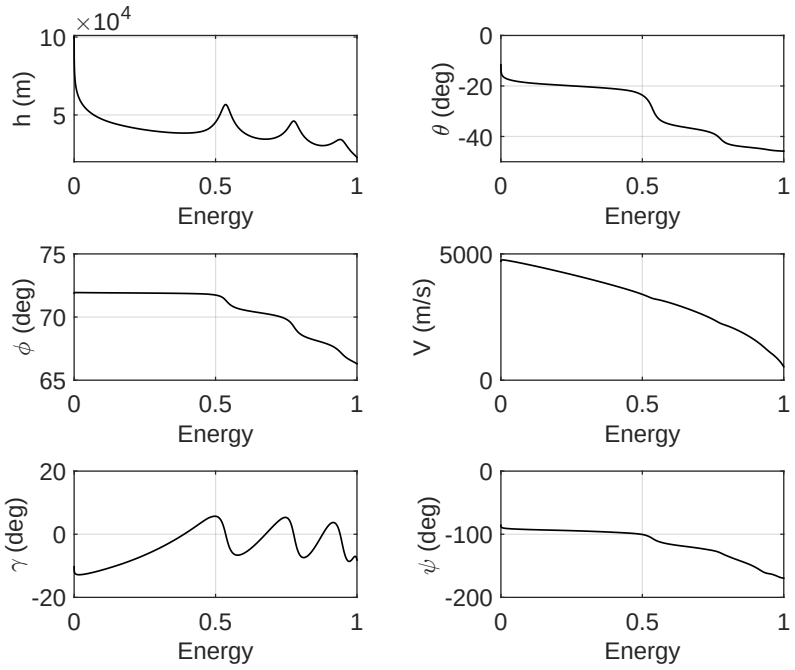


Figure 10.3: Nominal scenario - states.

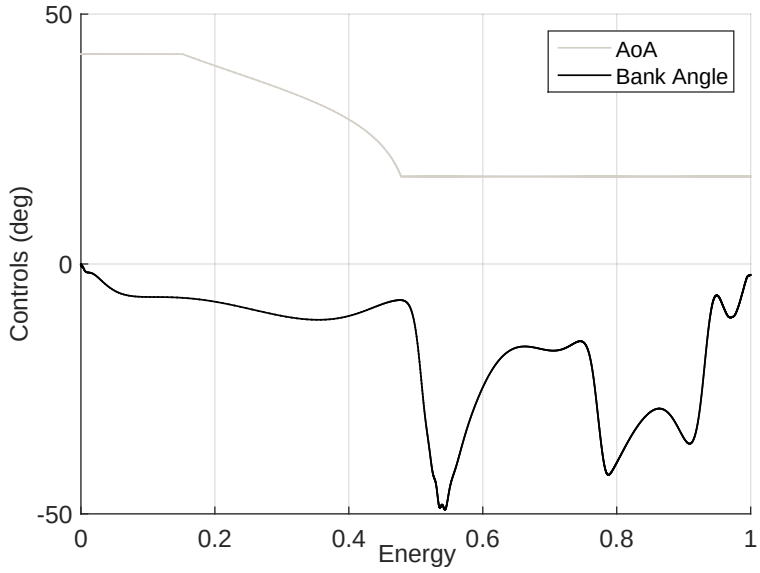


Figure 10.4: Nominal scenario - controls.

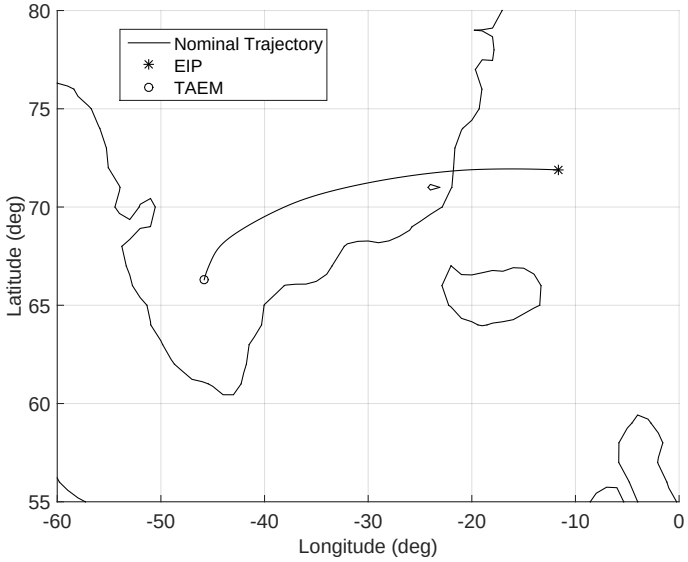


Figure 10.5: Nominal scenario - groundtrack.

second-order system, that is

$$\Delta D'' + 2\zeta_D \omega_D \Delta D' + \omega_D^2 \Delta D + k_D^i \int_{E_0}^{E_F} \Delta D dE = 0 \tag{10.43}$$

If we solve for the control u, we will obtain

$$u = \frac{D''_{ref} - a_D - 2\zeta_D \omega_D \Delta D' - \omega_D^2 \Delta D - k_D^i \int_{E_0}^{E_F} \Delta D dE}{b_D} \tag{10.44}$$

If we combine Eqs. (10.39), (10.42), and (10.44) we obtain the implemented backup guidance scheme for SHEFEX-3.

$$u = u_{ref} + \frac{-2\zeta_D \omega_D \Delta D' - \omega_D^2 \Delta D - k_D^i \int_{E_0}^{E_F} \Delta D dE}{b_D} \tag{10.45}$$

The terms a_D, and b_D can be computed with Eq. (10.37). The damping ratio ζ_D and the pseudofrequency ω_D are chosen as a compromise between the saturation of the controls, and the tolerated errors, while the integral term is included to improve the steady-state response of the system. The term pseudofrequency refers to the fact that we are not really dealing with a frequency (having units in Hz), but with something which is only a frequency from the mathematical point of vew, but is physically not, as the independent domain is energy. Therefore, the pseudofrequency is a parameter expressed in s^2/m^2. Moreover, since the energy domain is decreasing, to make the system stable, we need to impose eigenvalues with positive real part. This implies that the damping ratio has to assume negative values. The lateral guidance is treated exactly as done in the main guidance described in Sec. 9.3.

10.5 Angle of attack aid

Given the limits coming from the scenario, the bank angle may be not sufficient to have satisfactory response. Therefore, for the backup guidance, variations of the angle of attack are included. The feedback law proposed for the angle of attack is similar to what has been proposed by Mease[7]. The variations are limited to ± 1 deg w.r.t. the reference angle-of-attack profile. This choice is due to the fact that larger variations made the assumptions used in Sec. 10.1 no longer valid, with as a practical consequence that systematic violations of the constraints, in particular in terms of dynamic pressure, were observed. So, we define

$$\Delta\alpha = -k_\alpha \text{sign}(C_{D,\alpha})\, \Delta D \tag{10.46}$$

with the gain k_α positive.

10.6 Nominal Solution

In Figs. 10.6 through 10.7, 10.8, and 10.9 the results obtained with the drag-energy scheme for nominal conditions are depicted. The continuous lines show the results obtained with the use of the backup guidance, while the dashed lines show the results obtained in open-loop.

Also in this case, open loop and closed loop provide almost the same results. Slight differences are observed in terms of the commanded bank angle for two reasons. In the first place, in the drag-energy inversion scheme Earth's rotation is neglected, while in the closed-loop simulations is not. In the second place, the drag-energy profile is approximated by cubic splines (for the reasons explained at the beginning of this chapter), which capture the profile well, but are not exact between the nodes. This is a constraint of the scenario, which does not allow to have simple drag-energy profiles, like in the case of Space Shuttle (e.g., constant drag-energy segments), and therefore, it is an approximation we have to accept.

Now that the back-up guidance has been explained it is possible to test the two systems with extensive Monte Carlo simulation campaigns. This will be the subject of Chap. 11.

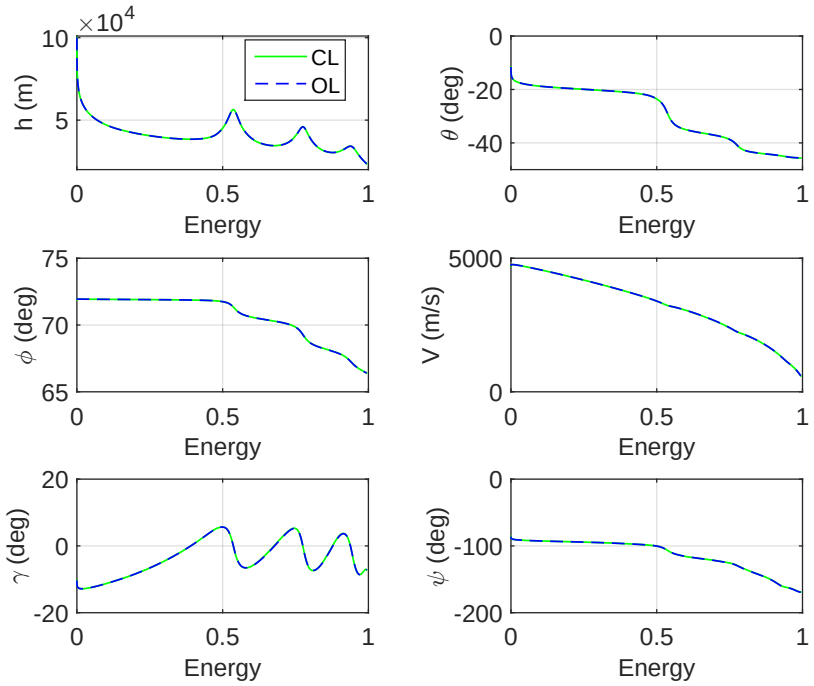


Figure 10.6: Nominal scenario - open loop vs closed loop: states.

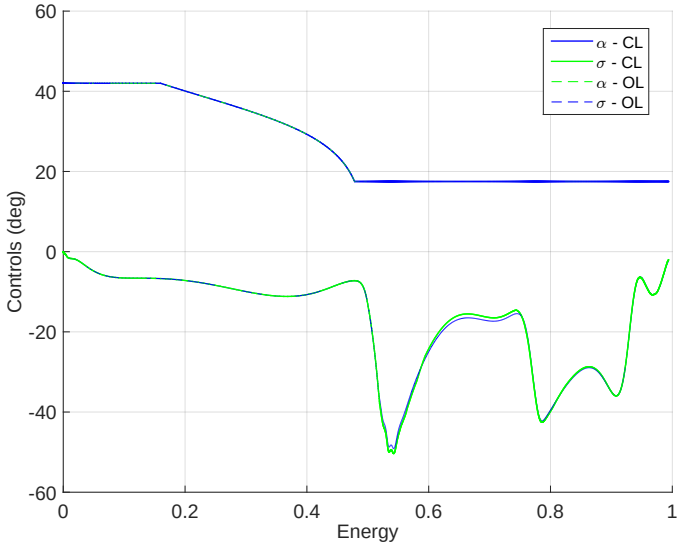


Figure 10.7: Nominal scenario - open loop vs closed loop: controls.

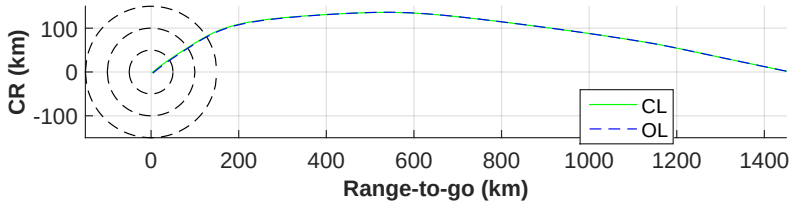


Figure 10.8: Nominal scenario - open loop vs closed loop: groundtrack.

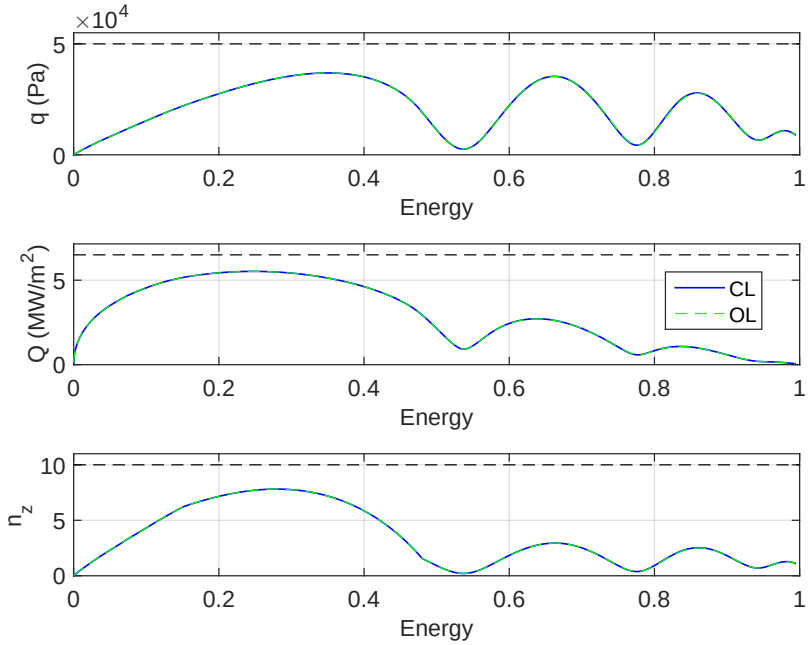


Figure 10.9: Nominal scenario - open loop vs closed loop: constraints.

Chapter 11

Monte Carlo Campaign

This chapter contains a description of the Monte Carlo campaigns performed to test the guidance systems developed. Uncertainties and disturbances are taken into account, and the results are summarized and commented. In Sec. 11.1 the parameters included in the Monte Carlo campaigns are described. The ranges included and the motivation for their choice are reported. Sections 11.2 and 11.3 report the results obtained for the main guidance, compared with open-loop simulations. In the first case, a perturbed US76 standard atmosphere has been used, while in the second case simulations have been performed by using a perturbed version of NRLMSISE-00. The backup guidance system has been tested as well, and results are reported and commented in Sec. 11.4 and 11.5, respectively. Also in this case, in the first case the perturbed US76 atmosphere has been used, while the perturbed NRLMSISE-00 has been included in the second campaign. Finally, Sec. 11.6 focuses on the comparison between the main guidance and the backup guidance. The comparison has been performed by using the results obtained with the perturbed US76 atmosphere; however differences in the behavior of the two guidance systems when NRLMSISE-00 is used are properly underlined in their respective sections, and in the conclusions of the work.

11.1 Monte Carlo Campaign - Parameters setup

In total the following effects have been included in the Monte Carlo campaign.

- Initial dispersions in terms of six states;
- Mass dispersion at the entry interface;
- Atmospheric density uncertainty;
- Lift coefficient uncertainty;
- Drag coefficient uncertainty;
- Wind profiles;

- F10.7 coefficients (only for NRLMSISE-00);
- Average F10.7 coefficients (only for NRLMSISE-00);
- Magnetic index data;
- Day and hour of the mission;

For the initial dispersions the values reported in Table 11.1 have been considered.

Table 11.1: Dispersions of initial states at the entry interface for the MC analysis.

State	Dispersions	Units
Geocentric Altitude	±250	m
Geocentric Longitude	±0.5	deg
Geocentric Latitude	±0.5	deg
Velocity Modulus	±70	m/s
Flight-path Angle	±0.5	deg
Velocity Azimuth Angle	±0.5	deg

The dispersion on altitude is the smallest w.r.t. the three initial components of position. It is assumed to be the value which "triggers" the guidance solution to be switched on, and for this reason, even a off-nominal but successfully flight will pass this point. For the latitude and the longitude, the range has been selected on the basis of the requirements reported in Table 2.4, and covers more than 5000 km^2 of uncertainty at the EIP altitude. For the velocity the uncertainty has been selected according to the launch and ascent specifications, which also provide the margins for the flight-path angle and the velocity azimuth angle. Both can assume variations up to ±0.5 deg w.r.t. their nominal values. This represents a sufficient margin, since it is more than what is usually taken into account in literature. For the mass we have a dispersion of ±0.5% w.r.t. the nominal mass of 500 kg. This value has been selected on the basis of ESA recommendation[23] for the residuals of propellant during the reentry. Indeed, ESA recommends to assume an uncertainty of 2% of the total mass of propellant. In case of SHEFEX-3 the total mass of propellant is 40 kg, resulting in 0.8 kg of uncertainty. To this margin, a further 1 kg may come from the uncertainty associated with the weighting operation of the vehicle before the launch, arriving up to 1.8 kg. To include further margin, 2.5 kg are considered.

For the atmospheric density, two simulation campaigns have been performed. In the former, the nominal US76 atmospheric density profile is perturbed up to ±20% w.r.t. its nominal value. The perturbation is time-varying, as it is a continuous functions of altitude. In the latter the more accurate NRLMSISE-00 has been considered, and perturbations up to 20% have been added as well. This has been done for the NRLMSISE-00 profile as well, which tends to have a reduced dispersion at lower altitudes. This agrees with what is usually done for vehicles such as the Space Shuttle and the IXV; in these cases balloons for density measurements at lower altitudes were used, having as a

result a reduction of the uncertainty for the atmospheric density from ground to 40 km. However, for SHEFEX-3 / ReFEX, the possibility to use these measurements is not certain. Therefore, the assumption of smaller dispersion at lower altitudes has not been included here[1]. For the case of NRLMSISE-00 several factors, like latitude, longitude, day of the year, and solar activity may modify the density profile. In the Monte Carlo campaign a nominal day has been considered, and variations up to ±30 days have been included. Moreover, variations of the coefficients F10.7, averaged F10.7 and magnetic indices, compatible with the available data[85,86] have been included. This corresponds to variations up to ±50% for all of these coefficients.

With this hypothesis we assume to know the period of the launch without knowing the exact day. However, in case a completely different period of the year is chosen as launch window, modifications to the nominal atmospheric model can be included, and a new analysis can be done according to the updated conditions with little modification to the software. The lift and drag coefficients can also be different w.r.t. their nominal values. Wind tunnels and numerical simulations of DLR aerodynamics experts[26] allow to suppose a variation up to 5% for both the coefficients, which can vary in an independent way, and are time-varying functions too (i.e., no simple bias is considered. Instead, the variations of coefficients continuously change with the angle of attack and the Mach number). Finally, wind profiles are included in the analysis too. The model taken into account is the aforementioned HWM07, which matches quite well with thousands of measurements done over the last fifty years[24]. Wind directly affects the aerodynamic coefficients, as they are function of the airspeed. In Fig. 11.1 some wind profiles have been plotted.

Table 11.2 summarized the described uncertainties. Once that the uncertainties and the

Table 11.2: Uncertainties for the MC analysis.

Uncertainty	Dispersions
$\Delta\rho$	±20%
ΔC_L	±5%
ΔC_D	±5%
Δm	±2.5 kg
$\Delta F10.7$	±50%
Average $\Delta F10.7$	±50%
ΔAph	±50%
Wind	HWM07

disturbances are defined, it is possible to report the results of the Monte-Carlo analysis. In total, 4 simulation campaigns have been performed.

[1]For testing purposes, simulations under this assumption were performed, having as a result a further improvement in the longitudinal and lateral dispersions, as expected.

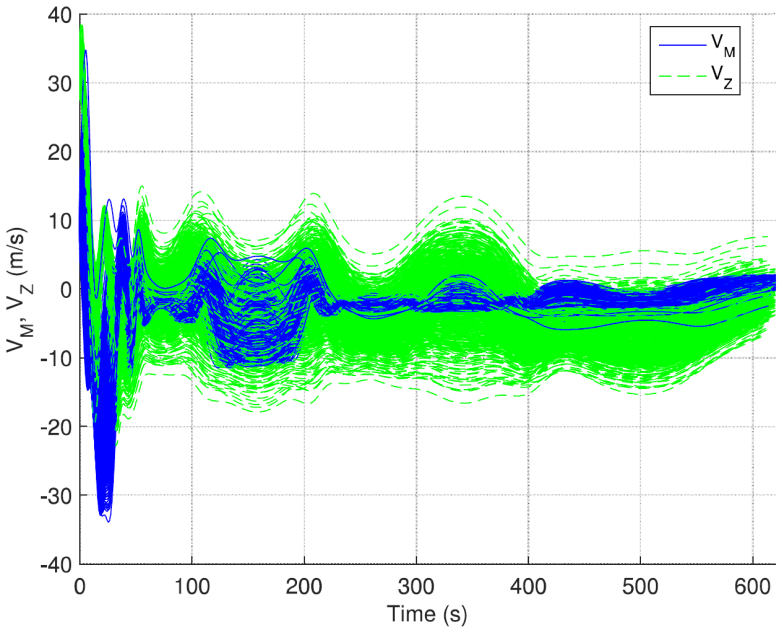


Figure 11.1: Horizontal Wind Model 07 profiles - meridional and zonal components.

11.2 Main guidance - Monte Carlo Campaign I

The first simulation campaign shows the performance of the main guidance system both in presence of initial states, uncertainties of atmospheric density (considering the perturbed US76 atmospheric model), mass, aerodynamics and wind. This will address the behavior of the overall main guidance. Figures. 11.2-11.15 show the results obtained for the main guidance.

With reference to Fig. 11.2 one can state that all the trajectories are stable, and smooth. The states envelope is within the prescribed limits, and so do the controls, plotted in Figs. 11.3 and 11.4. It is interesting to notice that, despite what happens in the backup guidance (in this case, as we will see, the angle of attack is only an aid; it is not integrated in the guidance scheme in systematic way, and therefore its use is limited, as it is reported in the MC analyses III, described in Sec. 11.4, and MC analysis IV, described in Sec. 11.5. In the main guidance instead, the angle of attack is fully integrated within the scheme, therefore larger variations of the angle of attack can be implemented without penalizing the performance of the system.

While upper values of the angle of attack (corresponding to upper C_L/C_D ratios) are hardly achieved, the lower saturation is achieved many times. This is due to the different ranges of $\Delta\alpha$ available w.r.t. the nominal value of 17.5 deg. Indeed, to increase the capability of the vehicle to counteract the uncertainties (e.g. a lower than expected C_L/C_D ratio, or an adverse wind profile), a variation up to +10 deg

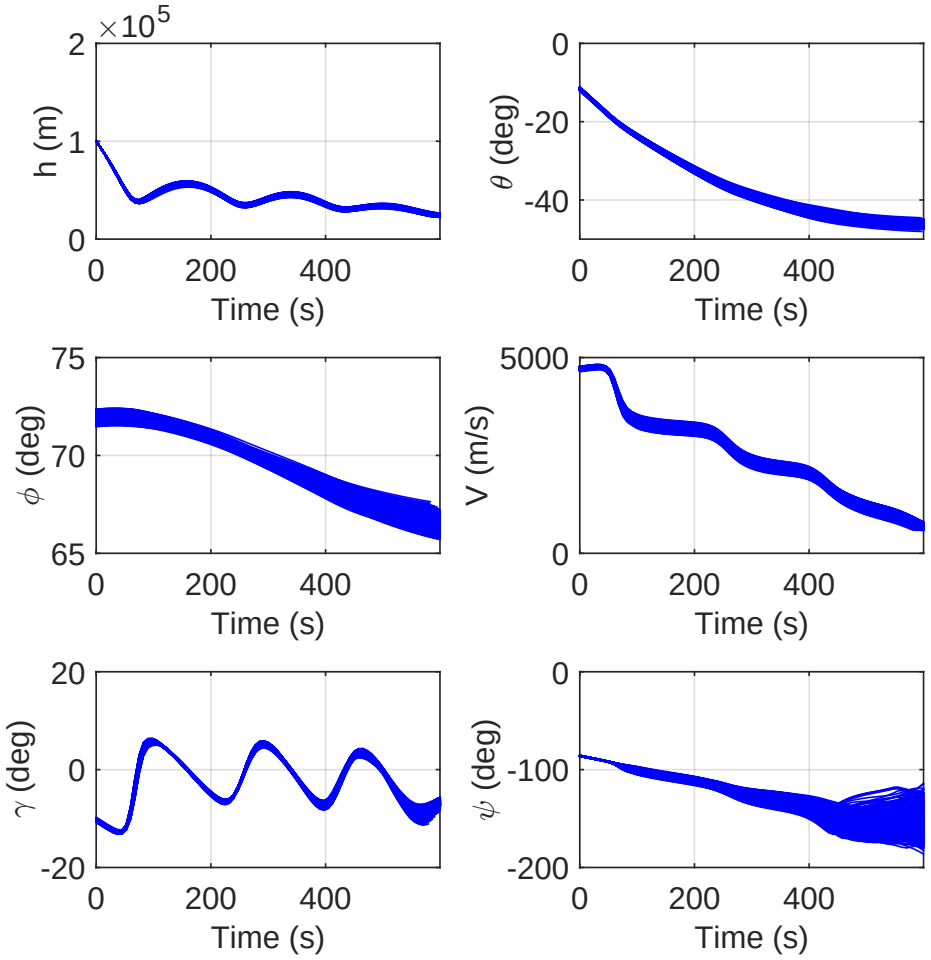


Figure 11.2: Monte Carlo campaign I - main guidance: states.

can be commanded. Reductions are instead limited to -2.5 deg, as the uncertainty of this particular database, beyond this region, increases. In terms of bank angle we can observe that, as expected, the maximum number of reversals is equal to 2. Except than in the first part (between 100 and 200 s) of some simulations, no saturation of the bank angle is achieved (absolute values of σ are always smaller than 60 deg). We can also see the two different minimum values the bank angle can achieve, as explained in Sec. 9.2.4, and which are -12.5 deg for negative bank angles, and +1 deg for positive bank angles. From Fig. 11.5 it is possible to observe how the initial direction of the vehicle is turned towards the prescribed terminal area, showing how the high-crossrange capability can be realized by the main guidance scheme, even in presence of strong uncertainties.

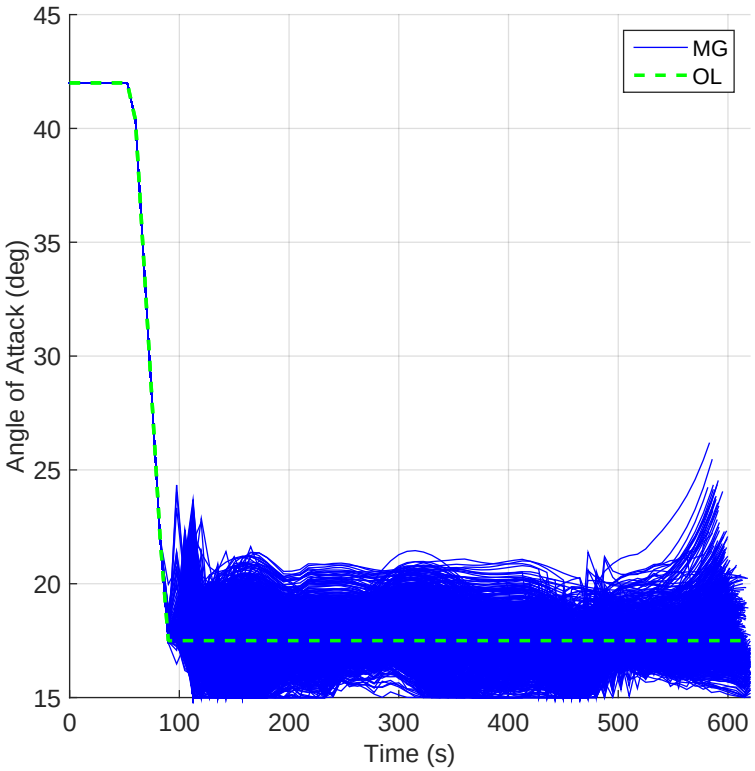


Figure 11.3: Monte Carlo campaign I - main guidance: angle of attack.

Let us compare the performance w.r.t. the open-loop trajectories. For both the cases, (closed loop with main guidance - MG, and open loop - OL), the simulations are stopped when the condition Mach = 2 is achieved. The dispersion areas are reported in Fig. 11.6, and summarized in Table 11.3, which reports how the cases are divided into the three circles defined as having radii equal to 50 km, 100 km, and 150 km.

The total dispersion area obtained with open-loop simulations is ≅34866 km^2. With closed-loop simulations it is reduced to ≅22078 km^2, resulting in an improvement of the dispersion area of about 36.7%. While in open loop only 46% of the cases fall into the smallest circle, this percentage is increased to more than 60% when the closed-loop

Table 11.3: Dispersion Distribution - 1000 cases.

Mode / Radius	50×50 km^2	100×100 km^2	150×150 km^2	outside
MG	605	340	55	0
OL	459	381	140	20

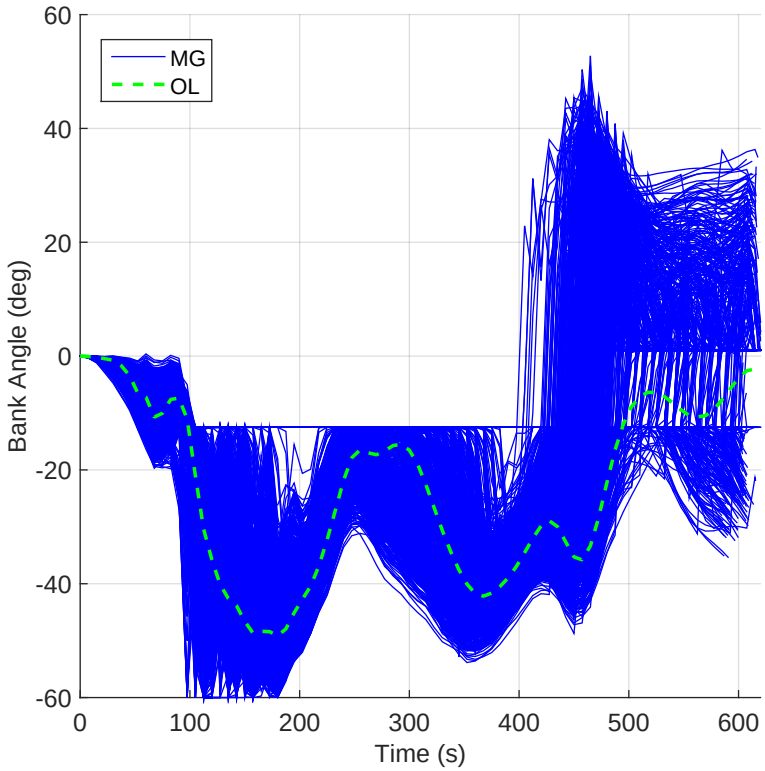


Figure 11.4: Monte Carlo campaign I - main guidance: bank angle.

guidance is used. Moreover, no cases fall outside in this case, against a 2% of cases, which, in open loop, do not satisfy the mission requirements. For the constraints, Figs. 11.7, 11.8, 11.9, and 11.10 show a comparison between the two Monte-Carlo campaigns in terms of peaks of dynamic pressure, heat flux, vertical load factor, and heat load, respectively. Specifically, no violations are observed in both cases, which means that in the scenario analysed the constraints are not the critical factor. For the dynamic pressure, the heat flux and vertical load factor the peaks are very similar as they all occur before the angle of attack modulation is completed, and therefore, before the feedback control law is used, and only the adaptive feedforward law is included.

For the maximum heat-load (Fig. 11.10), which is instead an integral factor, the benefits of using the main guidance are well visible. The maximum values are all associated with the open-loop cases, while most of 70% of cases in closed loop are in the range 1,350 - 1,470 MJ/m^2, against values up to 1,650 MJ/m^2 achieved in open loop. This aspect can be relevant in the design of the thermal protection system, and can potentially turn in a save in its mass, which brings an equivalent extra-mass for the experimental payload.

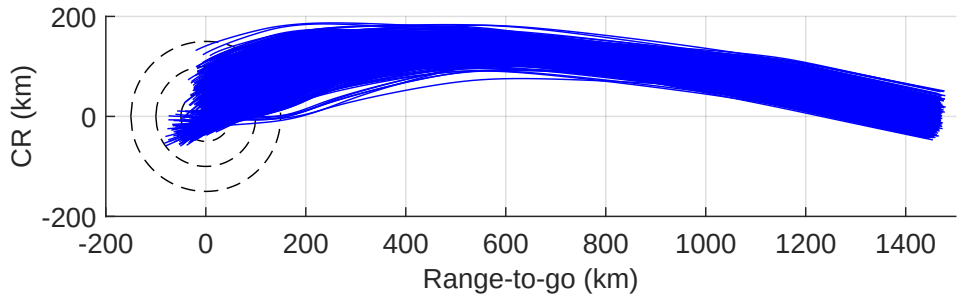


Figure 11.5: Monte Carlo campaign I - main guidance: groundtracks in DCA coordinates.

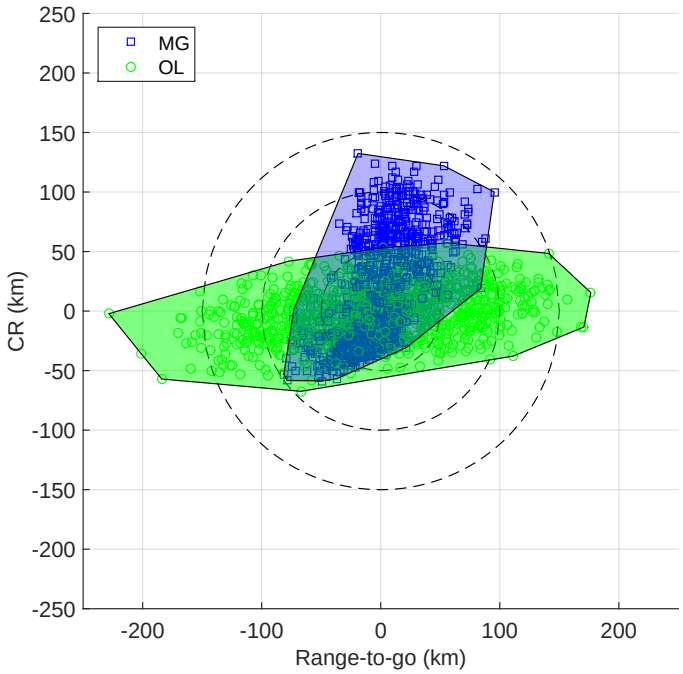


Figure 11.6: Monte Carlo campaign I - main guidance vs open loop: dispersion area in DCA coordinates.

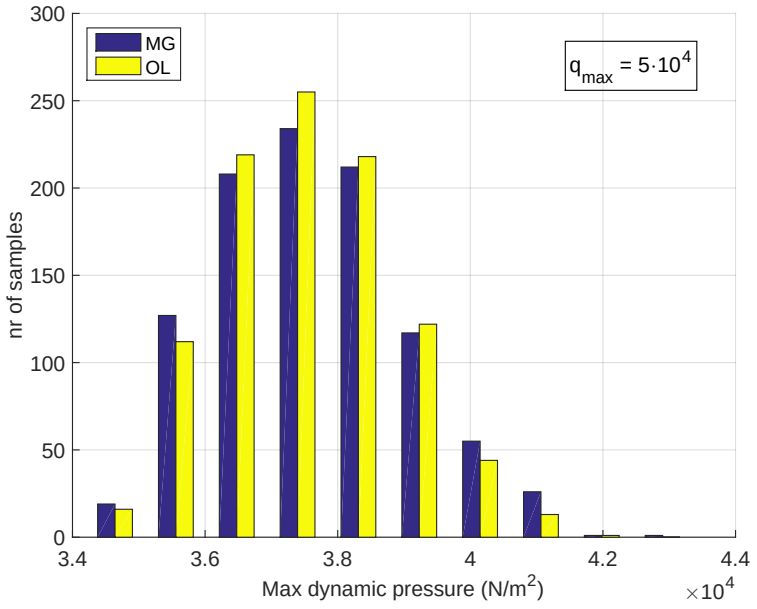


Figure 11.7: Monte-Carlo campaign I - main guidance vs open loop: dynamic pressure peaks.

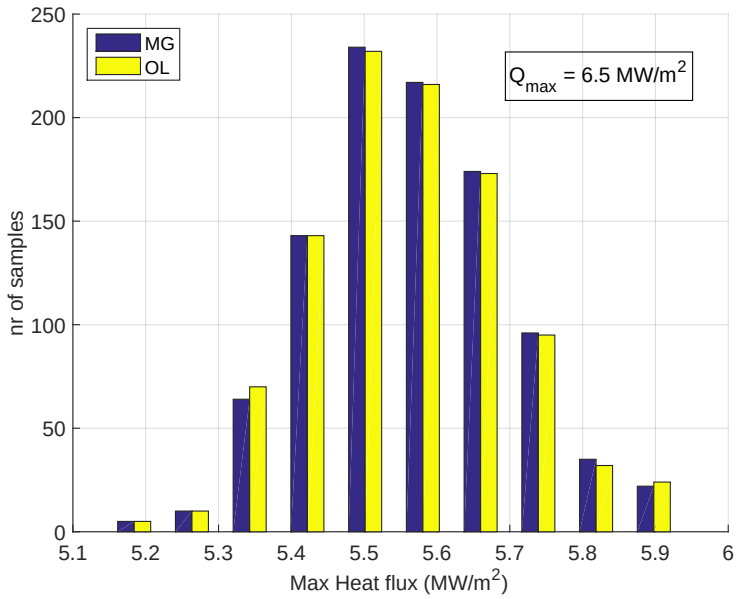


Figure 11.8: Monte-Carlo campaign I - main guidance vs open loop: heat flux peaks.

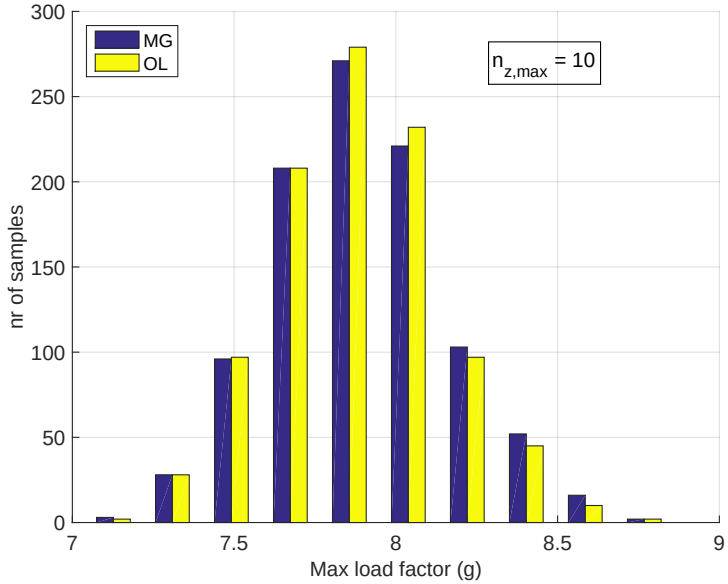


Figure 11.9: Monte-Carlo campaign I - main guidance vs open loop: vertical load factor peaks.

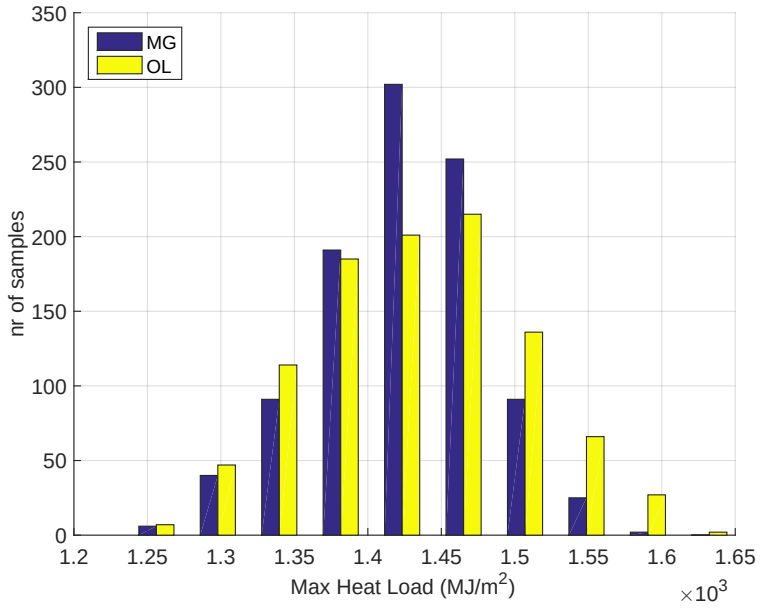


Figure 11.10: Monte-Carlo campaign I - main guidance vs open loop: heat load peaks.

In terms of longitudinal dispersions, the interface at Mach = 2 provides a natural interface for both the cases. Indeed, since the Mach number is a function of altitude and velocity, this condition will automatically limit their maximum dispersion. However, the improvements coming from the main guidance must be interpreted not in terms of longitudinal dispersion at Mach 2. Instead, we have to focus on how the lateral dispersions are improved once that the condition Mach = 2 is achieved. In other words, the closer to the prescribed final position the vehicle is when Mach = 2 is achieved, the better the behavior of the system is. From the comparison depicted in Figs. 11.11, 11.12, and 11.13 we can see the benefits of the use of the combined adaptive feedforward / robust-feedback controller proposed here.

While on altitude and velocity, the dispersions are similar, in terms of range-to-go, the adaptive feedforward guidance will improve the lateral performance, and the feedback controller will be able to bring the vehicle closer to the terminal position at the moment that the Mach = 2 condition is achieved. For instance, 574 cases fall within a distance w.r.t. the nominal target point of less than 46.5 km against 418 cases in open loop.

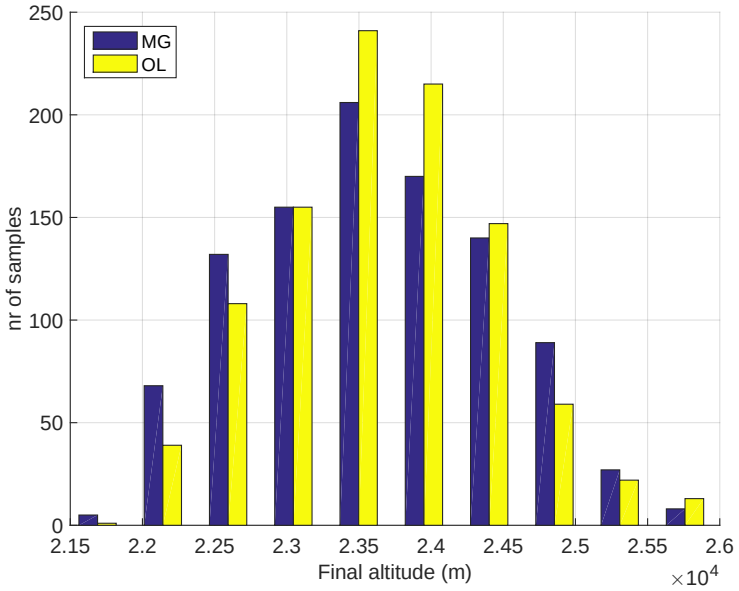


Figure 11.11: Monte-Carlo campaign I - main guidance vs open loop: final dispersions on altitude.

Finally, w.r.t. Figs. 11.14 and 11.15 we can observe that all the limits in terms of altitude and velocity are satisfied as well. Specifically, Fig. 11.14 show the envelope of flight on the altitude-velocity plane. Here as well we can observe that the trajectories are consistent and smooth. The TAEM area is depicted in Fig. 11.15. For this specific Monte-Carlo campaign no cases fall outside the red rectangle, representing the allowed area for the final velocity and altitude.

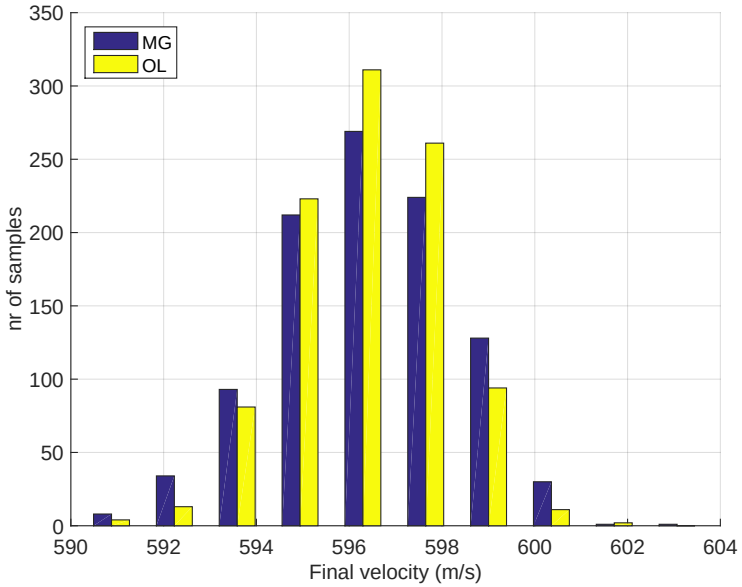


Figure 11.12: Monte-Carlo campaign I - main guidance vs open loop: final dispersions on velocity.

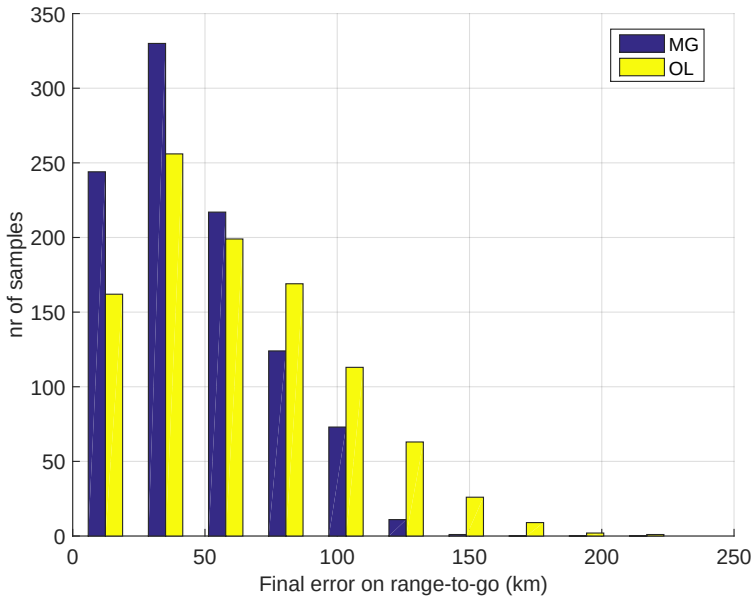


Figure 11.13: Monte-Carlo campaign I - main guidance vs open loop: final dispersions on range-to-go.

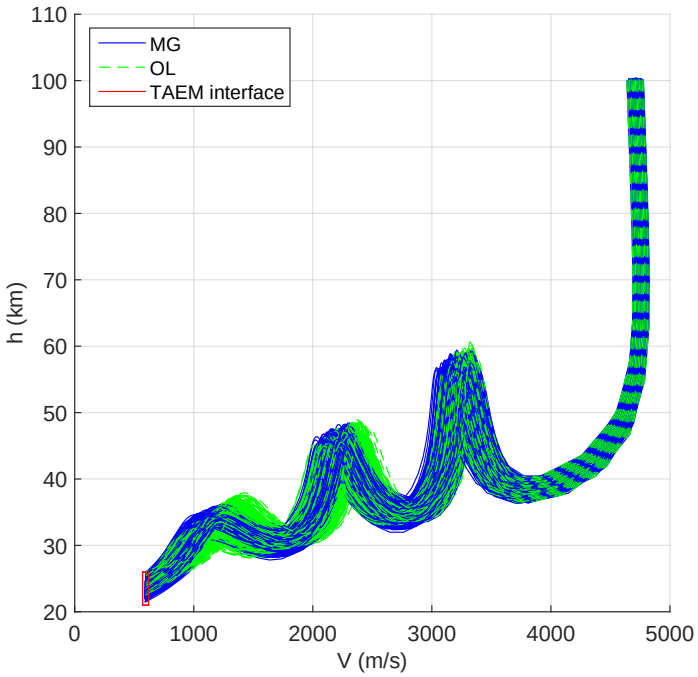


Figure 11.14: Monte-Carlo campaign I - main guidance vs open loop: h-V plane.

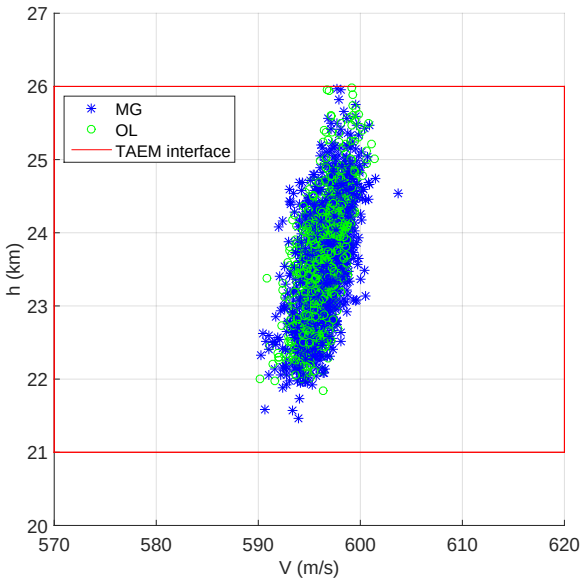


Figure 11.15: Monte-Carlo campaign I - main guidance vs open loop: h-V interface.

11.3 Main guidance - Monte Carlo Campaign II

The second simulation campaign shows the performance of the main guidance system both in presence of uncertainties of atmospheric density (considering the NRLMSISE-00 atmospheric model), mass, aerodynamics and wind. This will address the behavior of the overall main guidance in presence of a more realistic atmospheric model. Also in

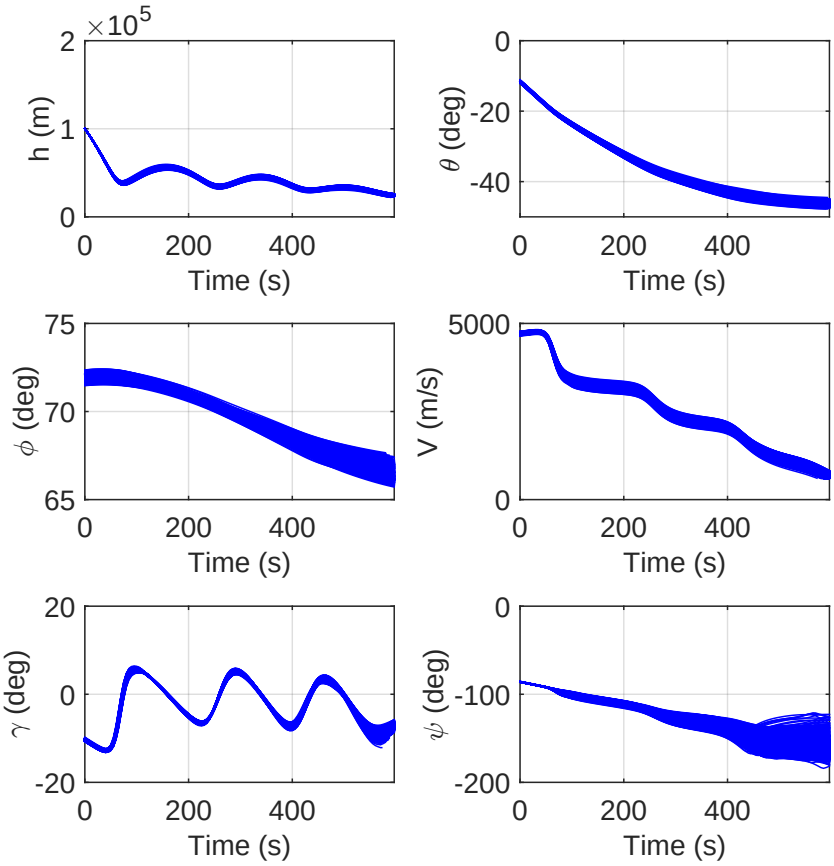


Figure 11.16: Monte-Carlo campaign II - main guidance: states.

this case, states and controls are smooth (Figs. 11.16-11.18), and all the trajectories show the high-crossrange capabilities achieved by the guidance algorithm (Fig. 11.19).

Let us compare the performance w.r.t. the open-loop trajectories. The overall areas covered in closed and open loop are slightly larger than in the previous case (OL - $\cong$ 35082 km^2, MG - 22121 km^2). This was expected as more variables are taken into account in this atmospheric model. The dispersion areas are depicted in Fig. 11.20 and some results are summarized in Table 11.4. Also in this case, Mach = 2 is considered as

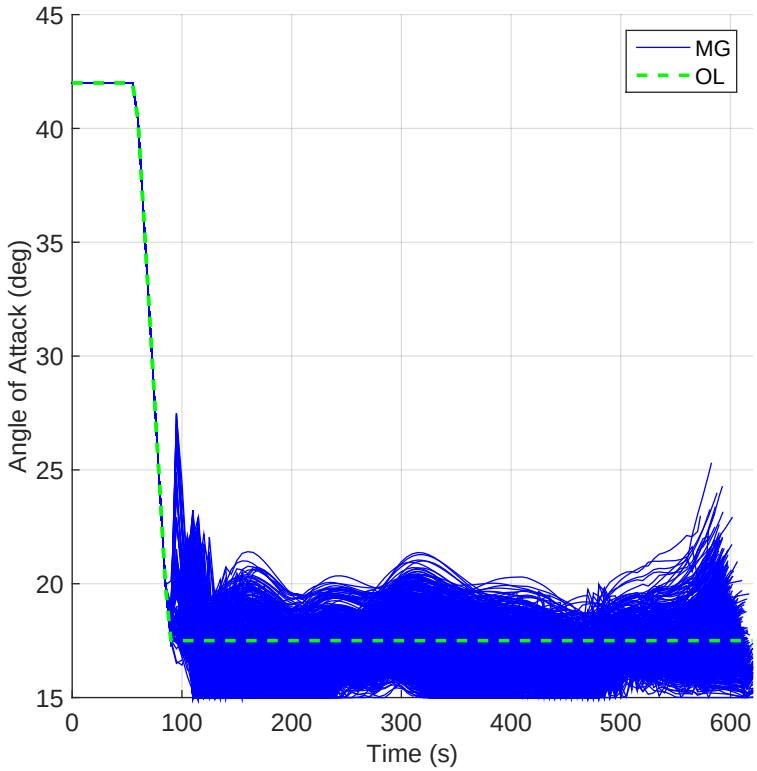


Figure 11.17: Monte-Carlo campaign II - main guidance: angle of attack.

terminal condition. Table 11.4 shows that, while in open loop only $\cong 45.1\%$ of the cases fall into the smallest circle, this percentage is increased to 56.5% when the closed-loop guidance is used.

In this case too, no cases fall outside in this case, against a 2.3% of cases, which do not satisfy the mission requirements in open loop. Let us compare the dynamic pressure (Fig. 11.21), the heat flux (Fig. 11.22), the vertical load factor (Fig. 11.23), and the heat load (Fig. 11.24) peaks. In terms of dynamic pressure, heat-flux, and load factor we can observe a wider distribution of ranges for each of these constraints when the main guidance is used. This is entirely due to the adaptive guidance, as the peaks

Table 11.4: Dispersion Distribution - 1000 cases.

Mode / Radius	50×50 km^2	100×100 km^2	150×150 km^2	outside
MG	565	362	73	0
OL	451	387	139	23

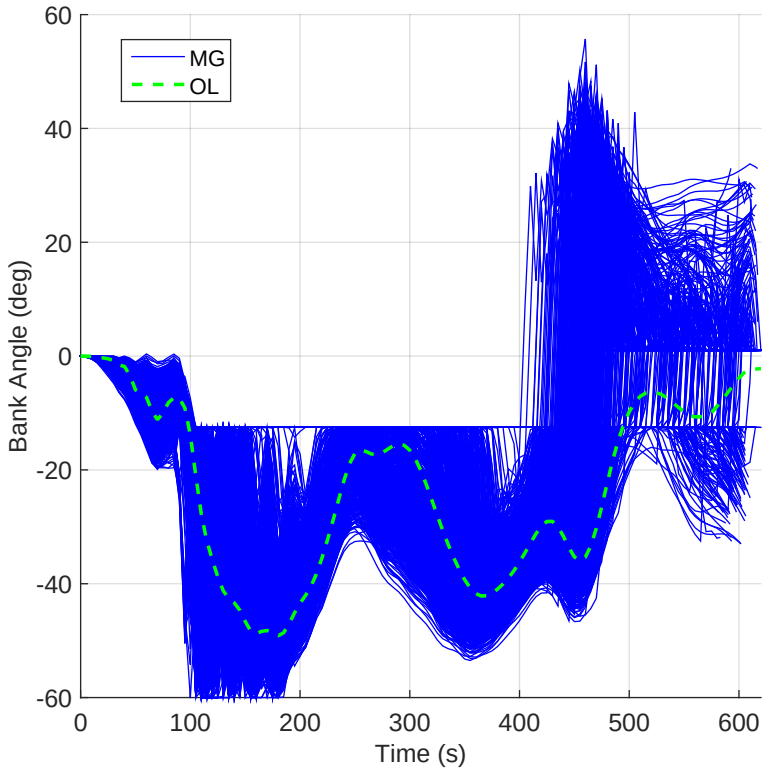


Figure 11.18: Monte-Carlo campaign II - main guidance: bank angle.

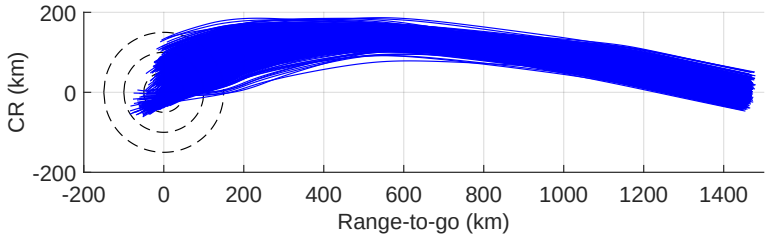


Figure 11.19: Monte-Carlo campaign II - main guidance: groundtracks in DCA coordinates.

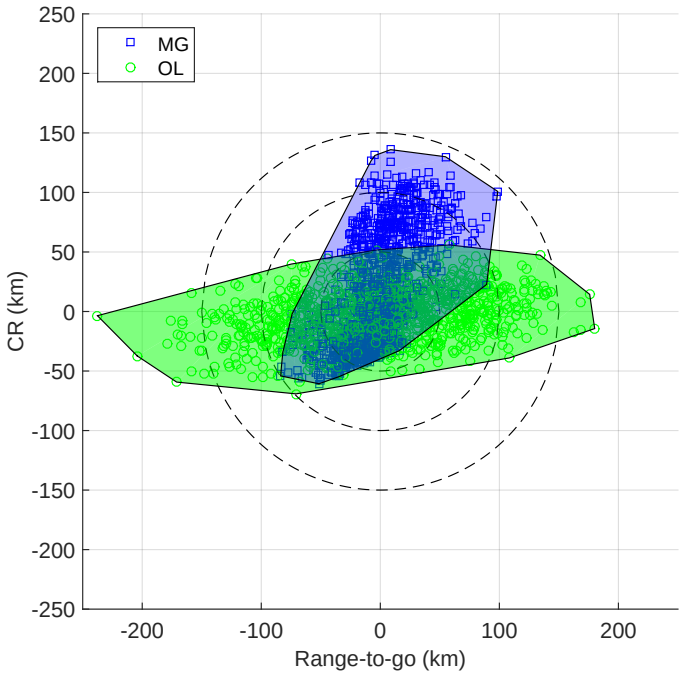


Figure 11.20: Monte-Carlo campaign II - main guidance vs open loop: dispersion area in DCA coordinates.

appear when the feedback controller has not been yet triggered. But why wider ranges? because the adaptive guidance does not take into account the constraints, and the combination of different off-nominal positions and velocities, combined with different bank angle profiles lead to a wider variations of conditions, while in open loop the same bank-angle profile is used. This aspect is stronger for the dynamic pressure, while it is less evident for the heat flux, and the vertical load factor.

Different considerations can be made for what regards the heat-load peaks (Fig. 11.24). As we said in the previous section, since this is an integral index, it can be associated with the main guidance. Indeed, in this case $\cong 80\%$ of the cases in closed loop have a maximum heat load, which is less than 1,500 MJ/m^2, and about 1% of the cases exceeds the value of 1,550 MJ/m^2, against more than 10% of the open-loop cases, which reach values larger than 1,550 MJ/m^2. The deductions made for what regards the possibility to relax the requirements, and therefore the mass of the TPS are confirmed by this MC campaign as well.

What happens in this case in terms of range-to-go? Also in this case, the Mach = 2 condition is reached in the two cases at a different distance w.r.t. the nominal conditions. For what regards the final altitude (Fig. 11.25), they tend to be similar, but slightly smaller than the ones associated with the open-loop cases. This is due to a higher

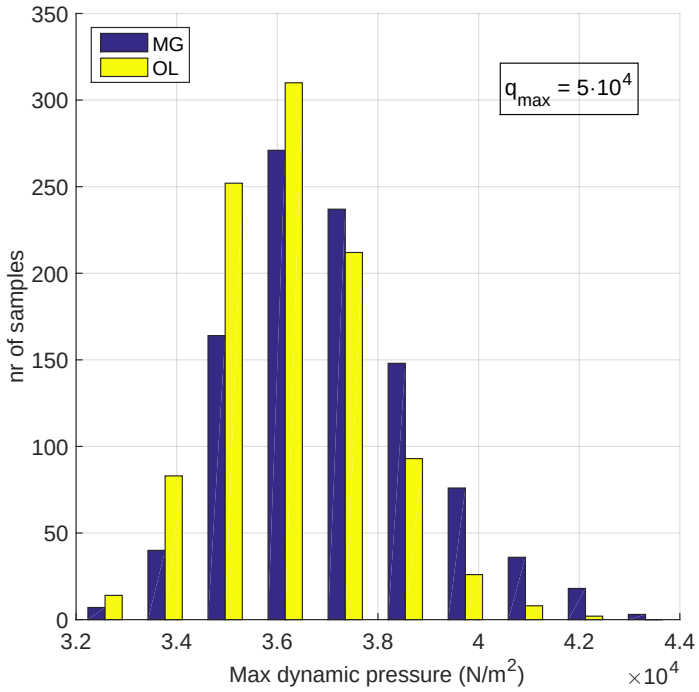


Figure 11.21: Monte-Carlo campaign II - main guidance vs open loop: dynamic pressure peaks.

control activity, which turns into a wider variation of the vertical component of lift. For this reason the velocity profiles look similar, but slightly higher than the open-loop cases, in virtue of the Mach = 2 condition (Fig. 11.26). In terms of range-to-go (Fig. 11.27), we can clearly observe the improvements brought by the use of the main guidance.

Specifically, 550 cases (associated with the main guidance) fall at a distance of less than 48 km w.r.t the target point, while this number is reduced to 445 when the open-loop cases are considered. Finally, the TAEM conditions on the $h - V$ plane are depicted as trajectories (Fig. 11.28) and final points (Fig. 11.29), respectively. All of the 1000 cases, except one, satisfy the requirements in terms of limits of altitude and velocity. However, since an acceptance criterion of 3σ is required, the main guidance fully satisfied these requirements.

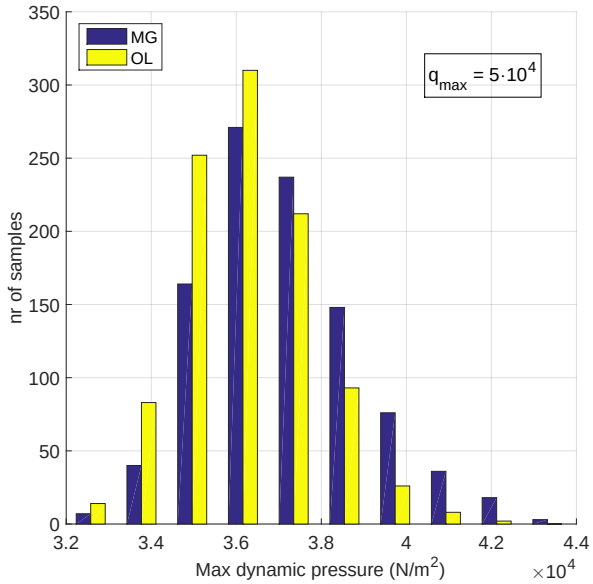


Figure 11.22: Monte-Carlo campaign II - main guidance vs open loop: heat flux peaks.

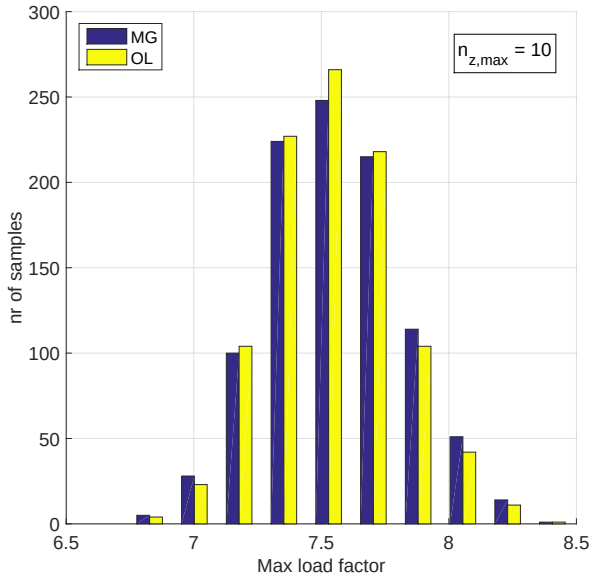


Figure 11.23: Monte-Carlo campaign II - main guidance vs open loop: vertical load factor peaks.

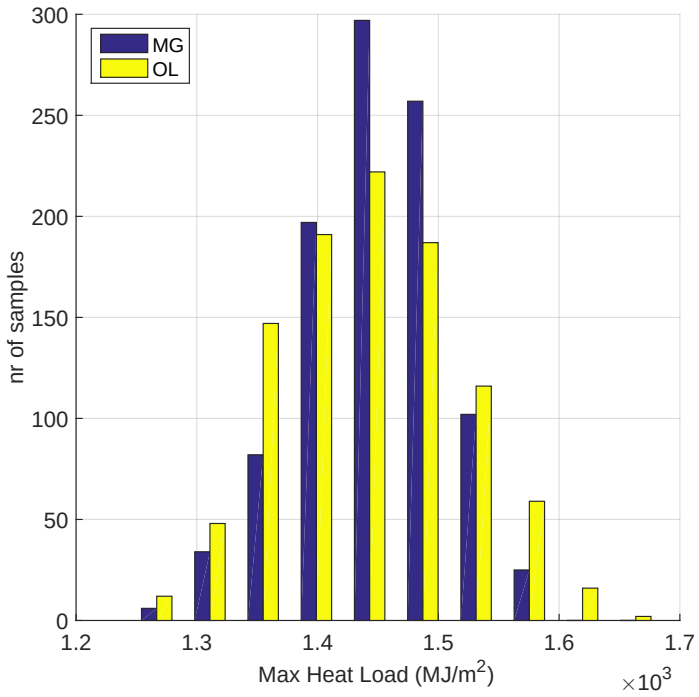


Figure 11.24: Monte-Carlo campaign II - main guidance vs open loop: heat load peaks.

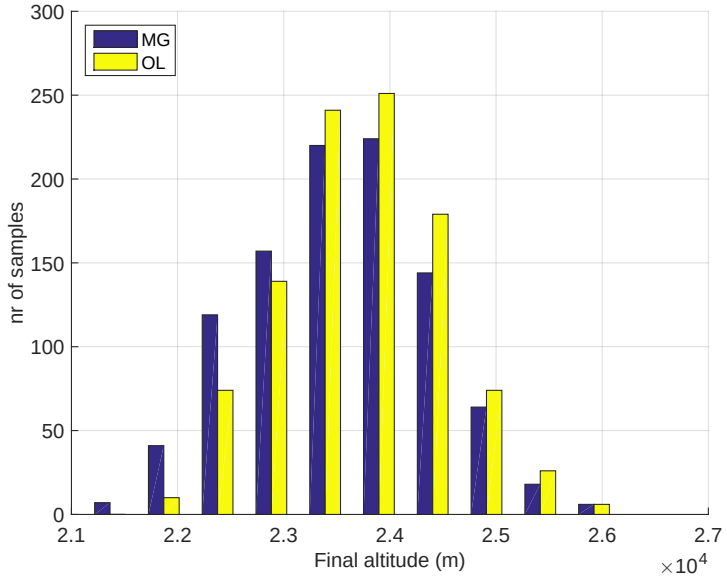


Figure 11.25: Monte-Carlo campaign II - main guidance vs open loop: final dispersions on altitude.

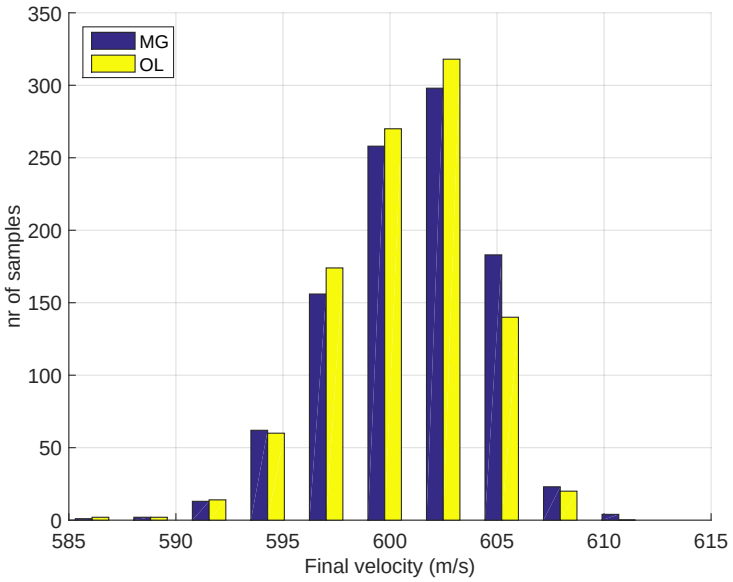


Figure 11.26: Monte-Carlo campaign II - main guidance vs open loop: final dispersions on velocity.

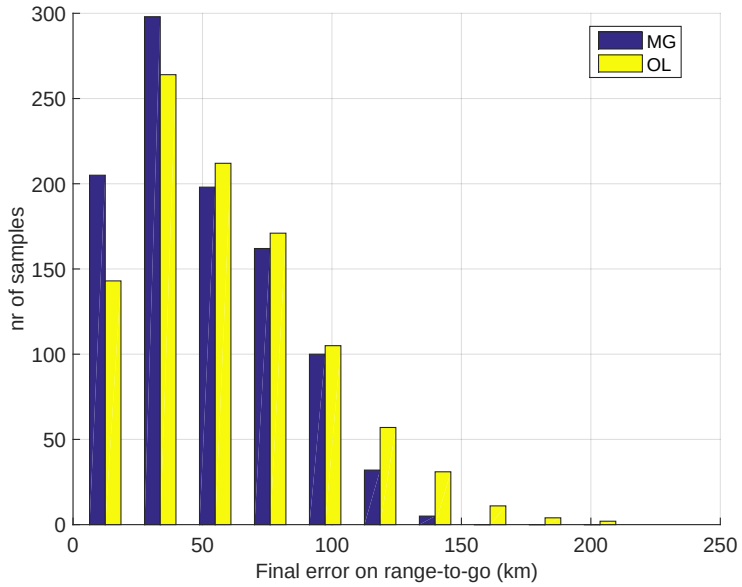


Figure 11.27: Monte-Carlo campaign II - main guidance vs open loop: final dispersions on range-to-go.

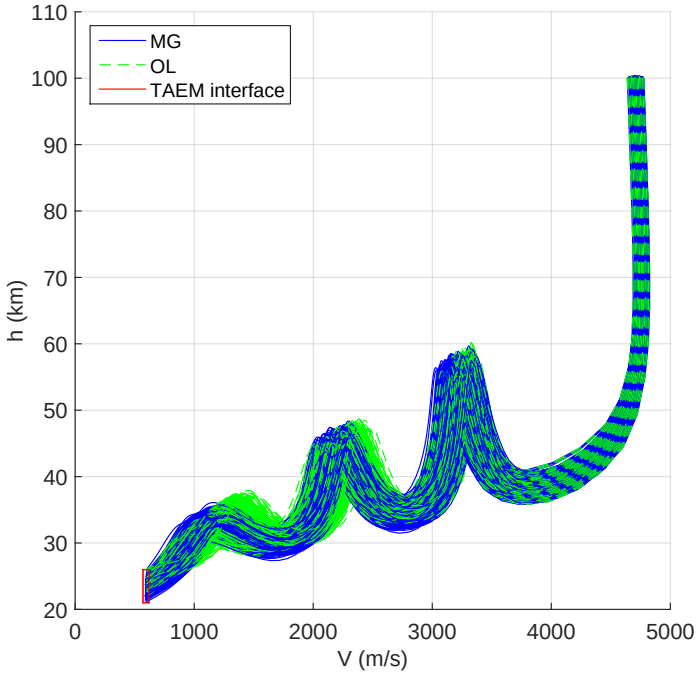


Figure 11.28: Monte-Carlo campaign II - main guidance vs open loop: h-V plane.

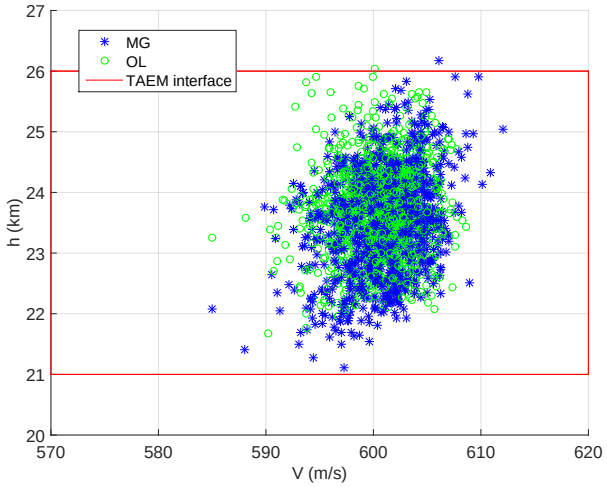


Figure 11.29: Monte-Carlo campaign II - main guidance vs open loop: h-V interface.

11.4 Backup guidance - Monte Carlo Campaign III

The third simulation campaign shows the performance of the backup guidance system in presence of uncertainties of atmospheric density (considering the perturbed US76 atmospheric model), mass, aerodynamics and wind. This will address the behavior of the backup guidance. For Monte Carlo campaigns III (Sec. 11.4) and IV (Sec. 11.5), the results of states and controls are showed w.r.t. the non-dimensional energy.

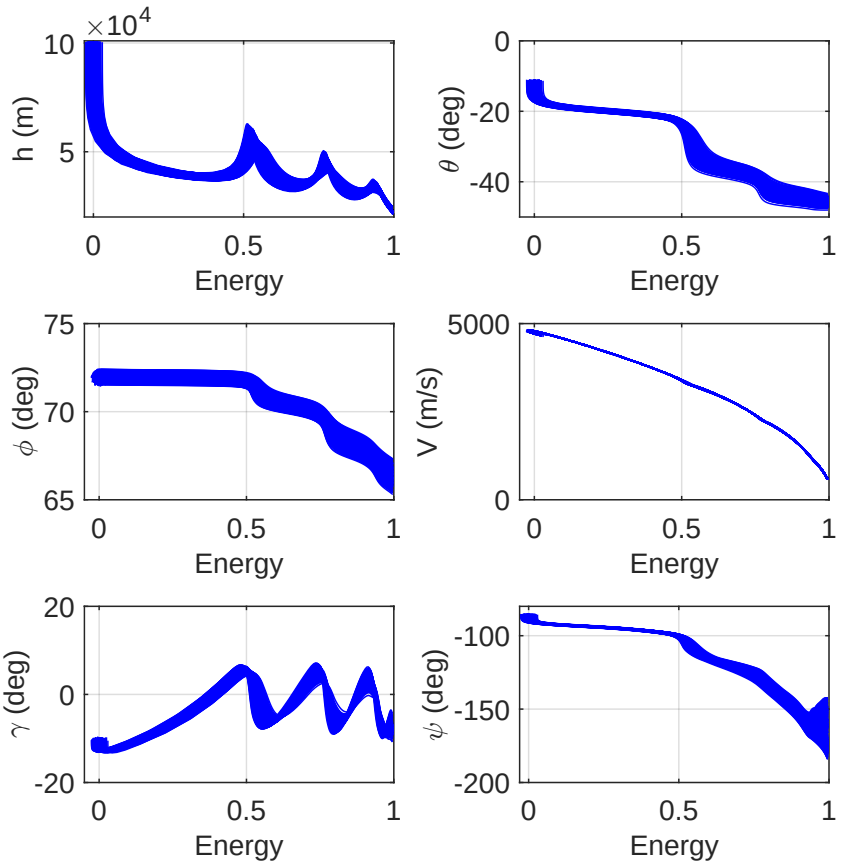


Figure 11.30: Monte-Carlo campaign III - backup guidance: states.

Figure 11.30 shows already a difference w.r.t. the main guidance. Indeed, the variations are slightly larger than the ones obtained with the MG, and the states are less smooth. This difference will become clearer in Sec. 11.6. As anticipated, the angle of attack is an aid, and therefore does not vary more than ±1 deg w.r.t. the nominal profile. Larger variations would cause a completely different drag dynamics, which would no longer be described well by Eqs. (10.13) - (10.36), as all the derivatives w.r.t.

Table 11.5: Dispersion Distribution - 1000 cases.

Mode / Radius	50×50 km^2	100×100 km^2	150×150 km^2	outside
BG	558	392	50	0
OL	459	381	140	20

α would be wrong.

In terms of bank angle, no significant variations are observed until values of nondimensional energy are less than $\cong 0.5$. This is due to the fact that this value corresponds to about 88 s, i.e., the triggering of the feedback control. The choice to introduce later the feedback term is not unique in literature[81,87], and is the best choice for this specific scenario. Since the open-loop controls are not scheduled w.r.t. energy, but w.r.t. time, the profile does not match with the ones associated with the use of the backup guidance. Moreover, it is worth to remind that the non-dimensional energy is computed according to Eq. (10.8), and theoretically goes from 0 (corresponding to the energy associated with the nominal entry conditions), to 1 (corresponding to the nominal TAEM). However, in the Monte-Carlo campaigns performed, the initial energy can be less than 0, (e.g. for a case associated with higher velocity and altitude. In those cases, the bank angle is kept equal to 0 until the non-dimensional energy value becomes equal to 0. This behavior is showed at the beginning of the profiles plotted in Fig. 11.32. The bank profile is provided as function of energy, but the guidance schemes run with a given frequency (10 Hz for all the cases). Since the energy at the beginning of the entry varies very quickly, in some cases the commands are sampled at energies, which can be slightly higher than 0. This aspect has no practical consequence on the results; indeed, they were crosschecked with simulations running at 100 Hz, and no significant variations in the results were observed.

In Fig. 11.33 the trajectories in DCA are plotted. All of them end in the prescribed limits. A difference w.r.t. the main guidance results is that in this case we see a wider distribution along the range-to-go direction, and a reduced distribution w.r.t. the crossrange direction. This is due to stricter ranges where the BG control acts, and can be observed better in Fig. 11.34.

Let us compare the performance w.r.t. the open-loop trajectories. Also in this case, the dispersion are is reduced w.r.t. the open-loop simulations, as expected (Fig. 11.34, Table 11.5). Also in this case we simulated 1000 cases. The total dispersion area obtained with open-loop simulations is $\cong$34866 km^2. With closed-loop simulations it is reduced to $\cong$22183 km^2, which is very close to the results obtained with the MG. Table 11.5 reports how the results are divided into circles having different radius. For this MC campaign 45.9% of the open-loop cases fall into the smallest circle, while this percentage is increased to 55.8% when the backup guidance is used. In this case too no cases (BG) falling outside the largest circle have been observed, against a 2% (OL) of cases, which do not satisfy the mission requirements.

Let us have a look at the peaks of the constraints, that is, the dynamic-pressure peaks (Fig. 11.35), the heat-flux peaks (Fig. 11.36), the vertical load-factor peaks (Fig.

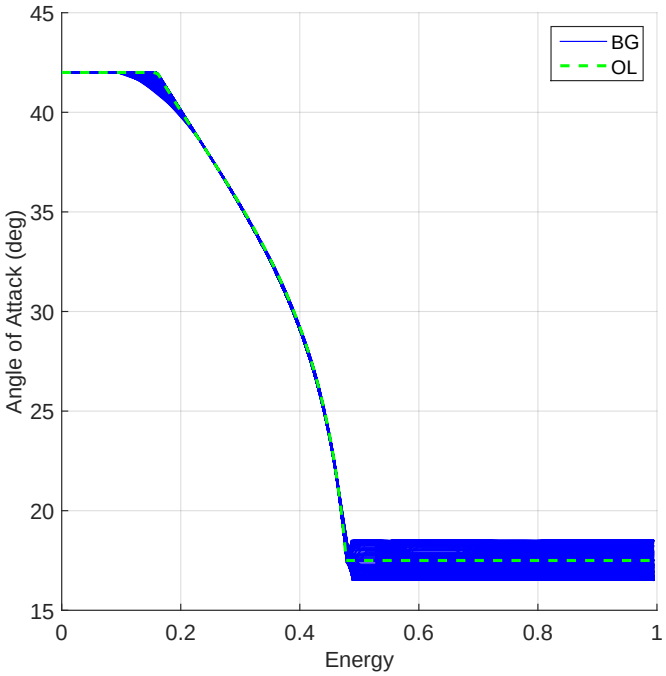


Figure 11.31: Monte-Carlo campaign III - backup guidance: angle of attack.

11.37), and the heat-load peaks (Fig. 11.38). In terms of dynamic pressure, we can see that the use of the backup guidance generates larger peaks, but all within the prescribed limits. This is due essentially to the cases where the range is shorter than the nominal one because of the uncertainties and disturbances.

In those cases, the guidance reduces the angle of attack to reduce the drag accordingly, and the lift decreases as well. Therefore, the velocity tends to be slightly larger than w.r.t. the nominal angle of attack case, and the altitude tends to be smaller. Moreover, this strategy has an effect on the drag dynamics, which would be perturbed by the different values that the derivatives w.r.t. α would assume. These effects generate meanly larger dynamic pressure peaks. For this reason the angle of attack aid is limited to ± 1 deg. Larger variations would amplify the effect described here, bringing violations in the limits of the dynamic pressure, while smaller values would bring to have final dispersions outside the prescribed limits reported in Table 2.4. These effects do not increase the maximum peaks of heat-flux (Fig. 11.36), and load factor (Fig. 11.37), where the backup guidance slightly improves the open-loop performance. Stronger improvements can be observed in terms of the heat-load maximum values (Fig. 11.38), where the use of the backup guidance strongly reduces the total heat experienced by the vehicle ($\cong$ 94% of the cases are below 1,470 MJ/m^2 against cases of 1,600-1,650 MJ/m^2 corresponding to the open-loop simulations.

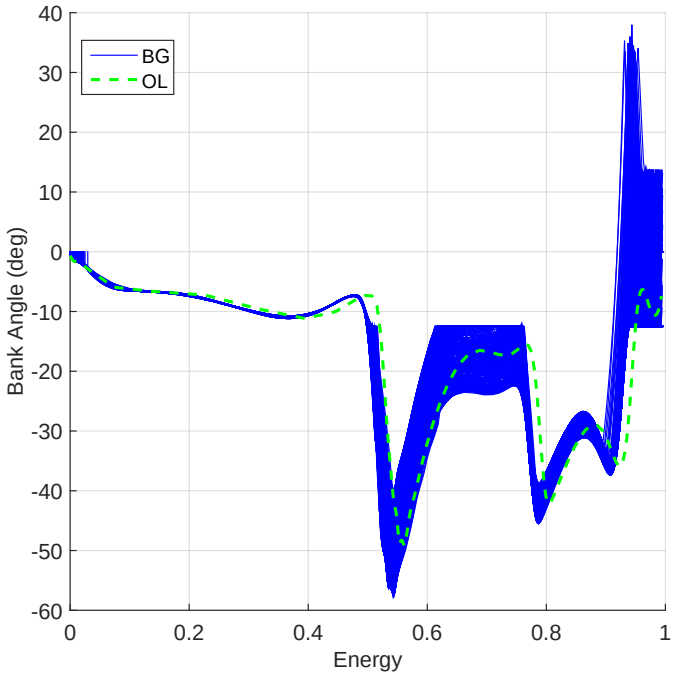


Figure 11.32: Monte-Carlo campaign III - backup guidance: bank angle.

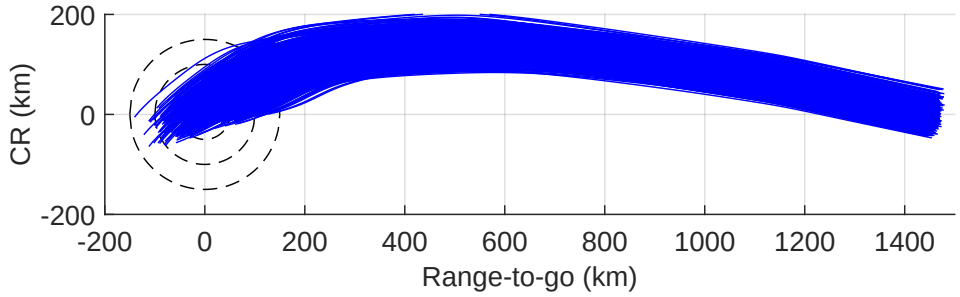


Figure 11.33: Monte-Carlo campaign III - backup guidance: groundtracks in DCA coordinates.

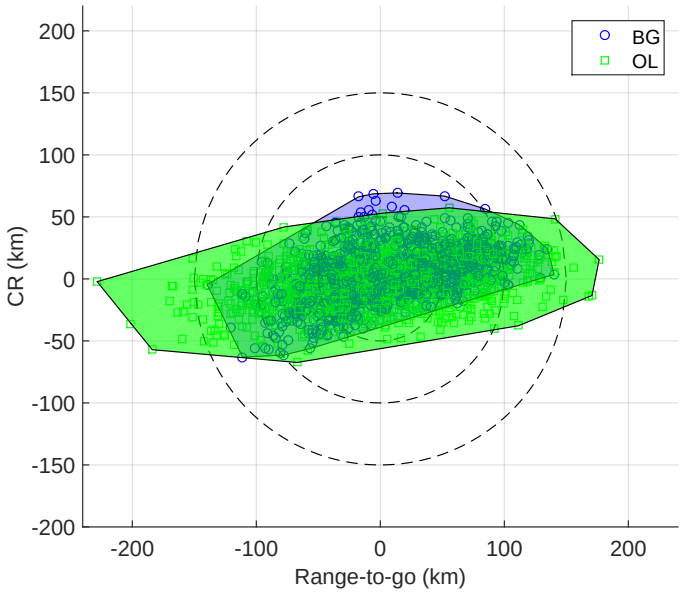


Figure 11.34: Monte-Carlo campaign III - backup guidance vs open loop: dispersion area in DCA coordinates.

As in the previous MC campaigns, the condition Mach = 2 indicates that no large dispersions w.r.t. the final condition on altitude (Fig. 11.39) and velocity (Fig. 11.40) are observed. However, the final range-to-go (Fig. 11.41) is, as expected, significantly improved by the use of the drag-tracking scheme. In this specific case 510 cases fall at a distance smaller than 46 km, against 416 cases in open loop. Figs. 11.42 and 11.69 show the corresponding profiles and TAEM conditions in the altitude-velocity plane. All the final values are well within the limits, but the trajectories are less smooth w.r.t. the main guidance.

Finally, Fig. 11.44 shows the drag-energy profiles in closed loop. It is possible to observe that all the profiles are within the prescribed limits, here represented as corresponding drag-acceleration profiles. However this is only an indicative information, as the limits do not take the uncertainties into account, and is formally valid only when the nominal models for the environment, the mass, and the aerodynamic are considered, in absence of wind.

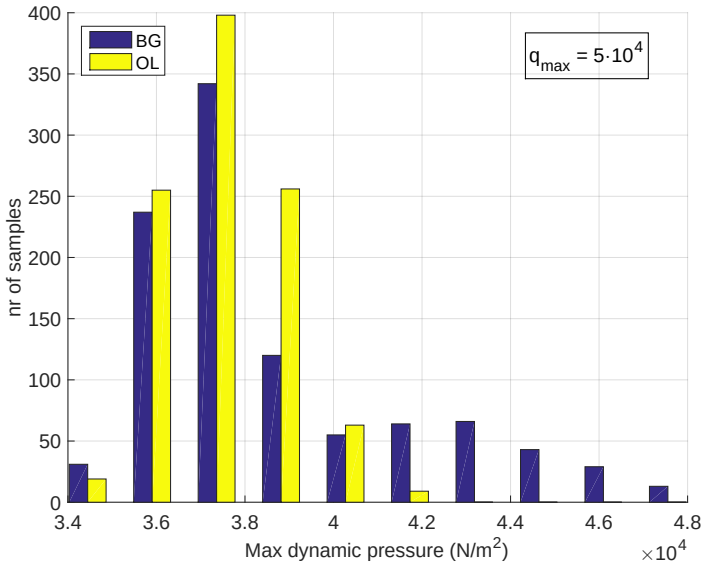


Figure 11.35: Monte-Carlo campaign III - backup guidance vs open loop: dynamic pressure peaks.

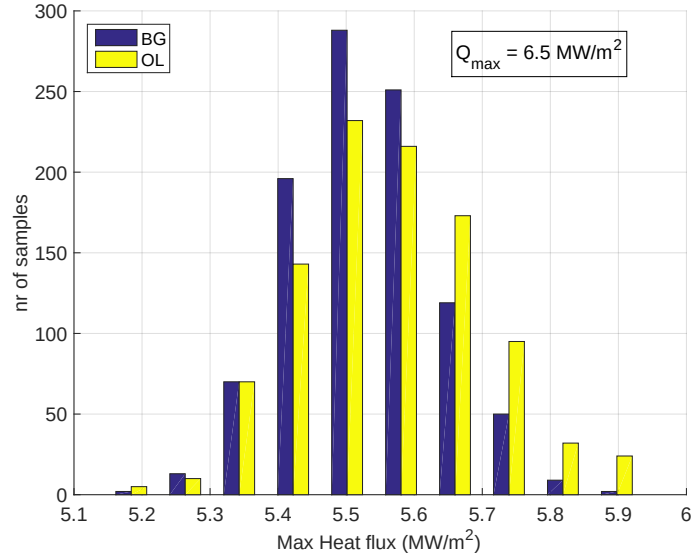


Figure 11.36: Monte-Carlo campaign III - backup guidance vs open loop: heat flux peaks.

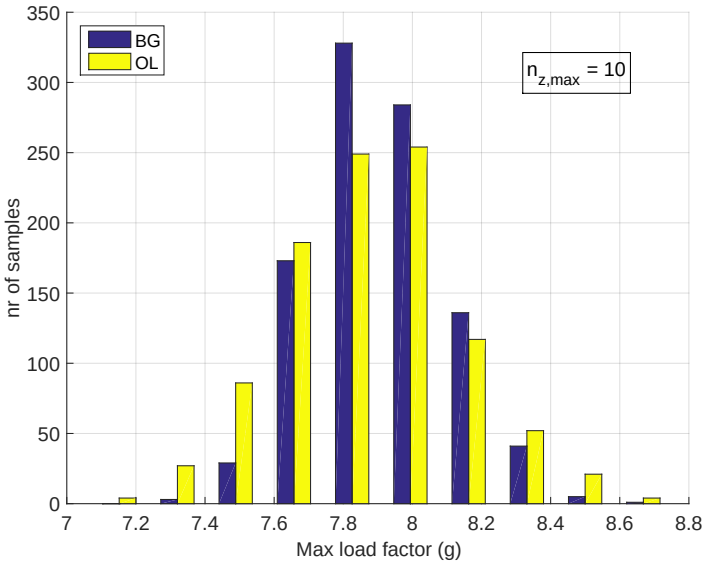


Figure 11.37: Monte-Carlo campaign III - backup guidance vs open loop: vertical load factor peaks.

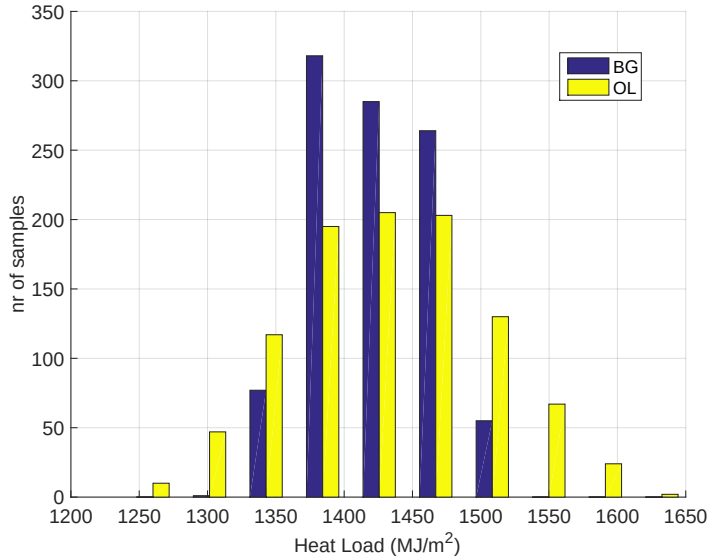


Figure 11.38: Monte-Carlo campaign III - backup guidance vs open loop: heat load peaks.

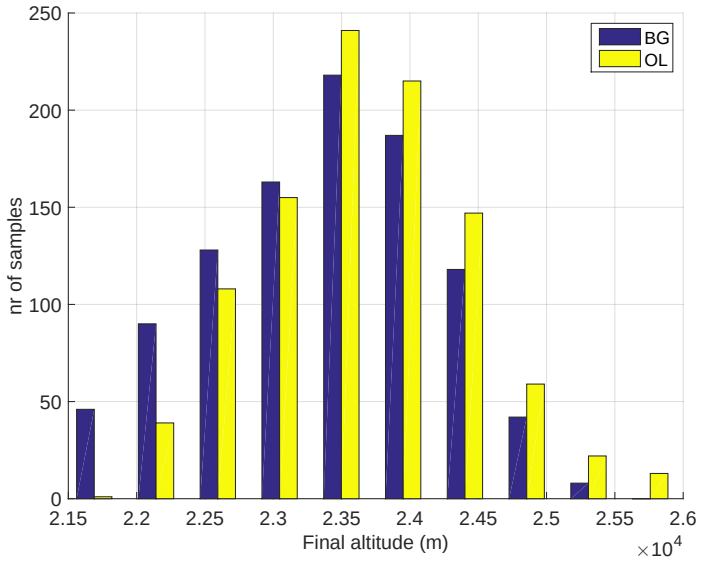


Figure 11.39: Monte-Carlo campaign III - backup guidance vs open loop: final dispersions on altitude.

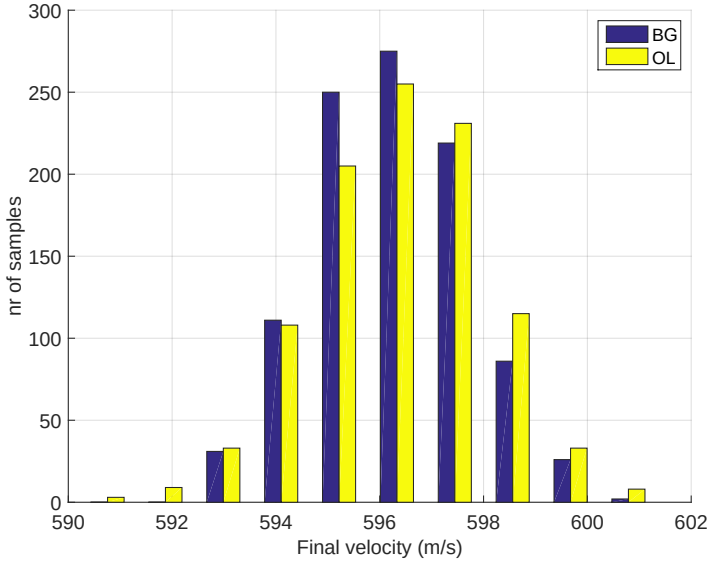


Figure 11.40: Monte-Carlo campaign III - backup guidance vs open loop: final dispersions on velocity.

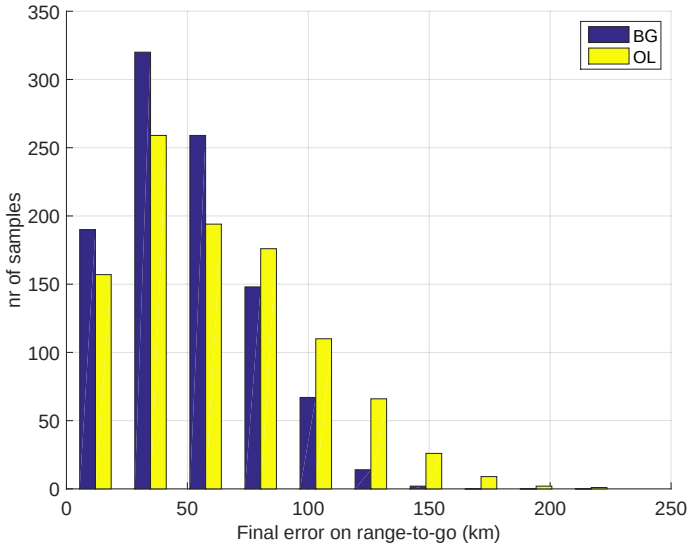


Figure 11.41: Monte-Carlo campaign III - backup guidance vs open loop: final dispersions on range-to-go.

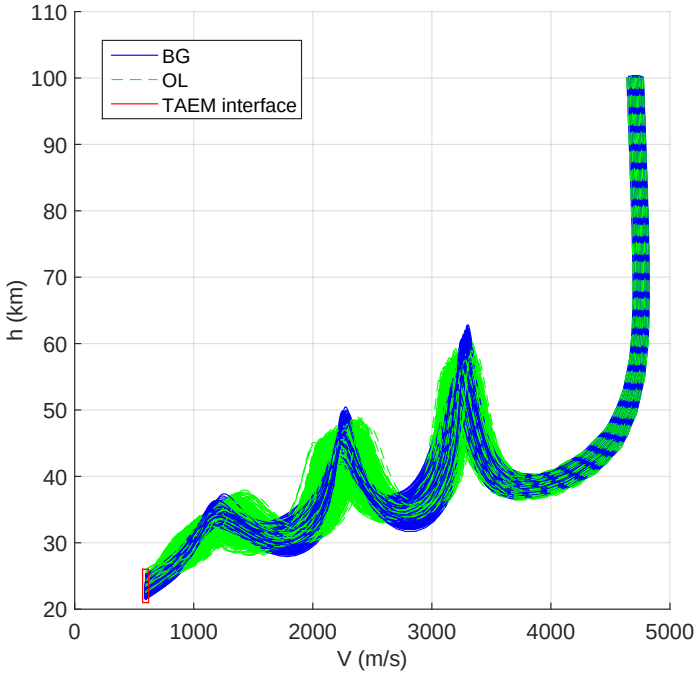


Figure 11.42: Monte-Carlo campaign III - backup guidance vs open loop: h-V plane.

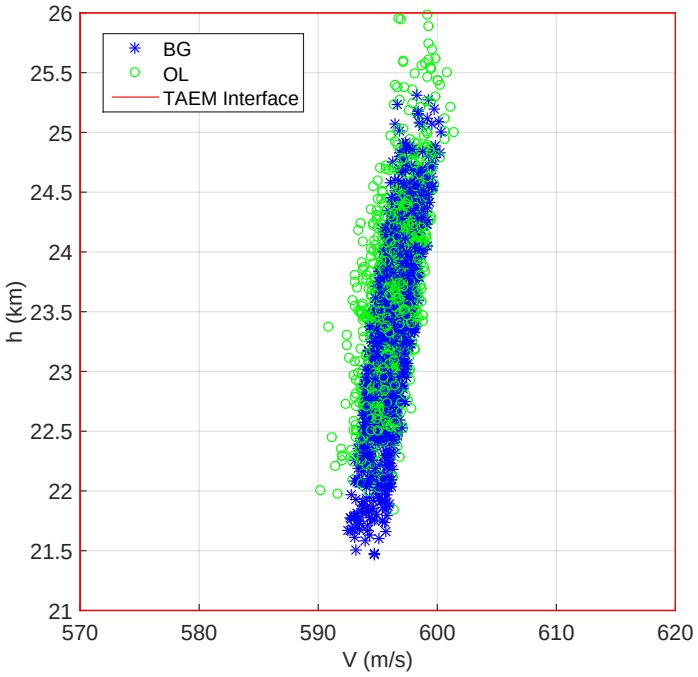


Figure 11.43: Monte-Carlo campaign III - backup guidance vs open loop: h-V interface.

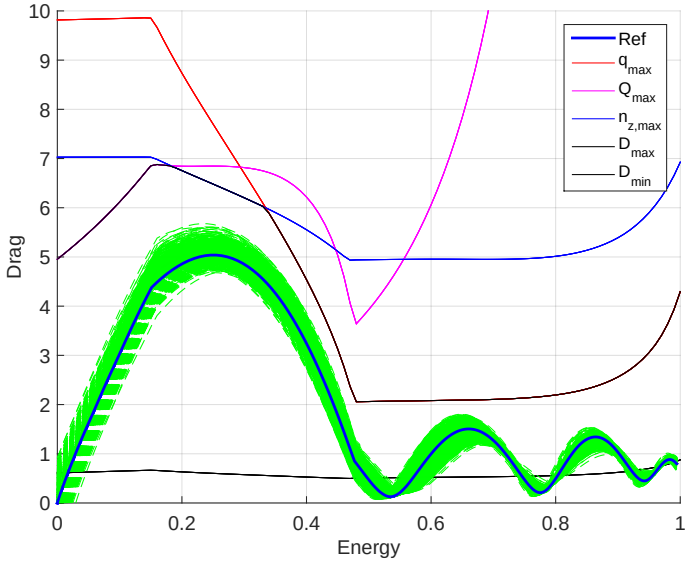


Figure 11.44: Monte-Carlo campaign III - backup guidance vs open loop: drag vs energy envelope.

11.5 Backup guidance - Monte Carlo Campaign IV

The fourth and last simulation campaign shows the performance of the backup guidance system both in presence of uncertainties of atmospheric density (considering the NRLMSISE-00 atmospheric model), mass, aerodynamics and wind. This will address the behavior of the overall backup guidance in presence of a more realistic atmospheric model.

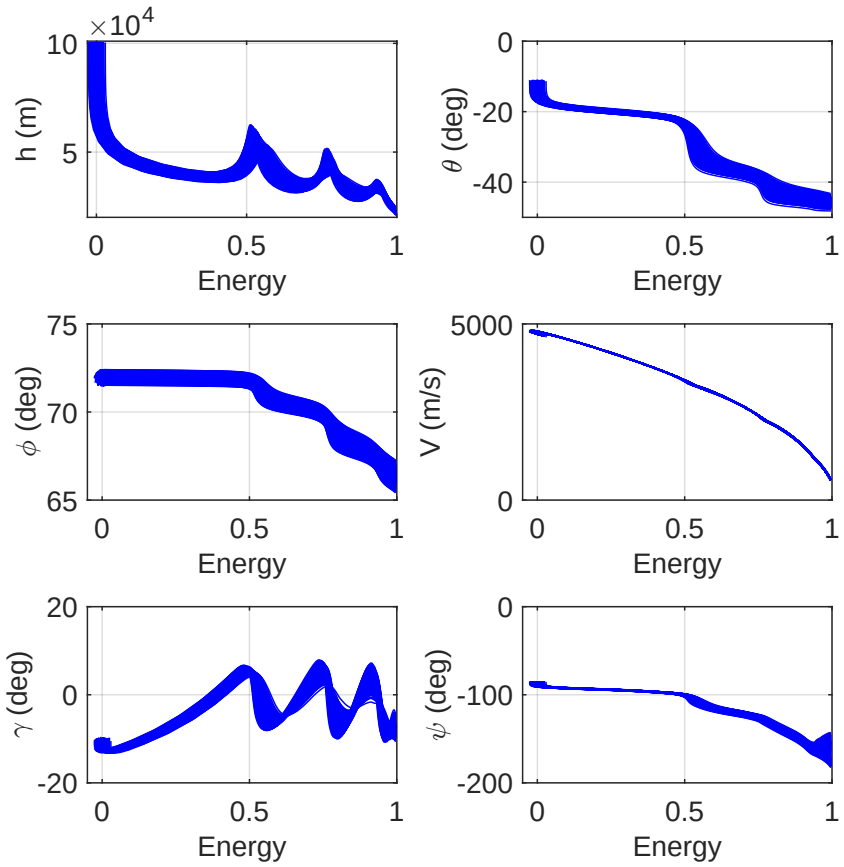


Figure 11.45: Monte-Carlo campaign IV - backup guidance: states.

Also in this case the states (Fig. 11.45), the controls (Figs. 11.46 and 11.47), and the trajectories (Fig. 11.48) fulfill all the requirements listed in Table 2.4. As done in the previous cases, let us compare the performance w.r.t. the corresponding open-loop trajectories. The total dispersion area obtained with open-loop simulations is $\cong$ 35082 km^2. With closed-loop simulations it is reduced to $\cong$ 19635 km^2, which is theoretically better than what provided by the MG.

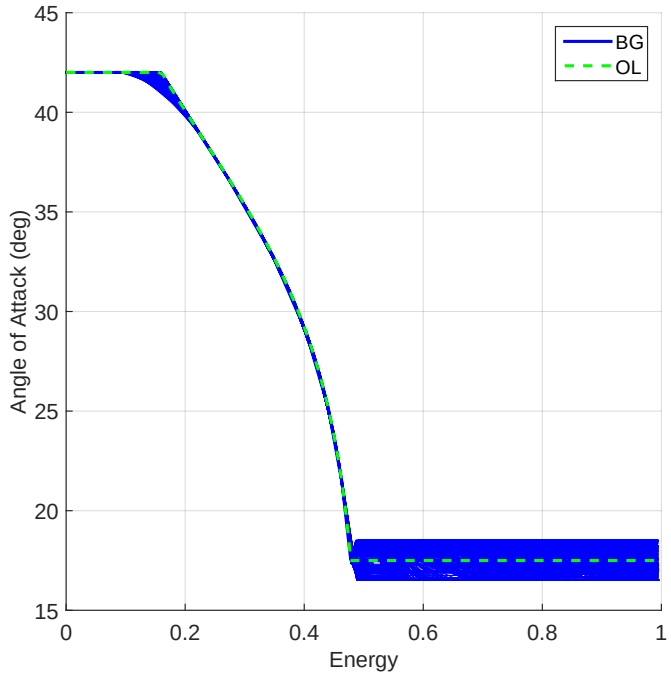


Figure 11.46: Monte-Carlo campaign IV - backup guidance: angle of attack.

Table 11.6: Dispersion Distribution - 1000 cases.

Mode / Radius	50×50 km^2	100×100 km^2	150×150 km^2	outside
BG	614	342	44	0
OL	451	387	139	23

While in open loop only 47.2% of the cases fall into the smallest circle, this percentage is increased to 61.4% when the closed-loop guidance is used. Moreover, no cases fall outside in this case, against 2.3% of cases, which do not satisfy the mission requirements in open loop. However, even if the performance are slightly better w.r.t. the results of the main guidance, in this case we can observe a violation in terms of peaks of dynamic pressure (Fig. 11.50); 1.6% of the cases violate the maximum dynamic pressure that the vehicle can experience, while for the main guidance this does not happen. This is due to the non-optimality of the drag-energy guidance strategy for an oscillating entry like the one proposed here.

In terms of heat flux (Fig. 11.51), and heat load, (Fig. 11.53), the backup guidance brings a significant reduction of the peaks. For instance, when the backup guidance is used, 91.6% of the cases are below 5.65 MW/m^2; this percentage drops to 79.5% in open loop. The maximum heat load experienced for the closed-loop cases is 1,550

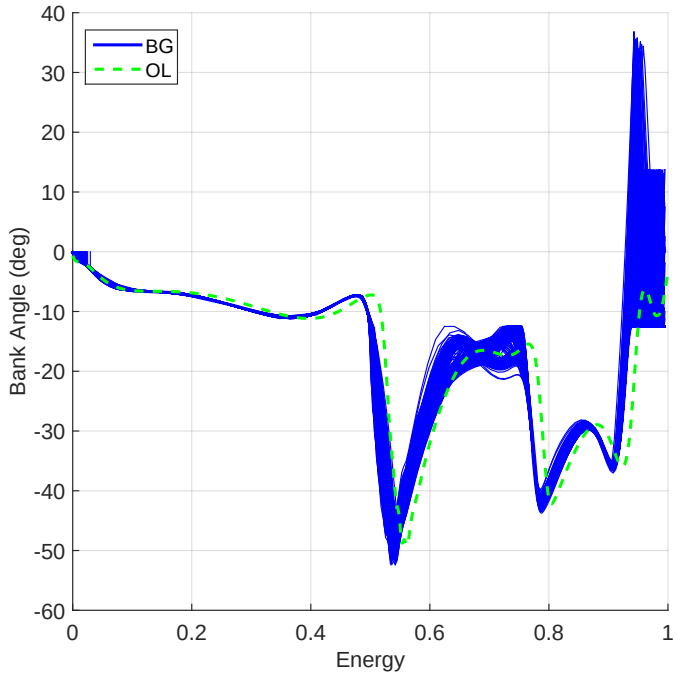


Figure 11.47: Monte-Carlo campaign IV - backup guidance: bank angle.

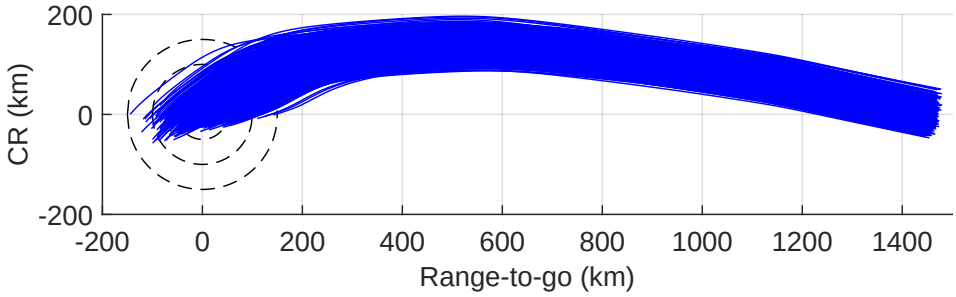


Figure 11.48: Monte-Carlo campaign IV - backup guidance: groundtracks in DCA coordinates.

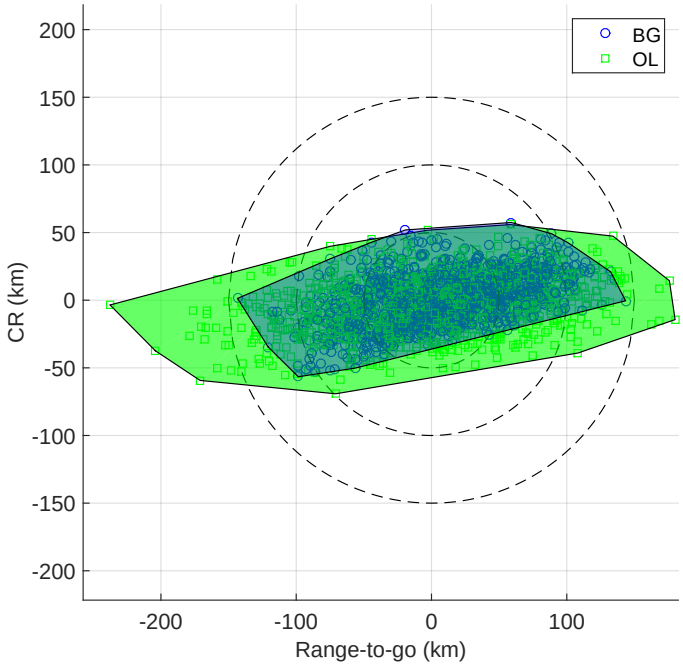


Figure 11.49: Monte-Carlo campaign IV - backup guidance vs open loop: dispersion area in DCA coordinates.

MJ/m^2, while in open loop the peak jumps to 1,660 MJ/m^2. No significant differences are observed in terms of load factor (Fig. 11.52). For the final dispersions in terms of altitude and velocity (Figs. 11.54, and 11.55) we can observe a lower range of altitudes counteracted by an upper range of velocities. For what regards the range-to-go (Fig. 11.56), 523 cases fall within a distance of less than 42 km w.r.t. the nominal TAEM, against 394 cases in open loop. Also in this case, the TAEM conditions on the altitude-velocity plane are depicted in Figs. 11.57 and 11.58.

As in the previous simulation campaign, in Fig. 11.59 the drag envelope obtained for the closed-loop simulations is depicted. No violations of the maximum drag appear here, and this seems to be in contradiction with the violation of the maximum dynamic pressure. However the limits depicted in Fig. 11.59 and the conditions associated with the violations are not the same, as the former are based on nominal models, while the latter on perturbed models. When the perturbed models are used to generate the drag, the violations become visible in the drag-energy plane as well.

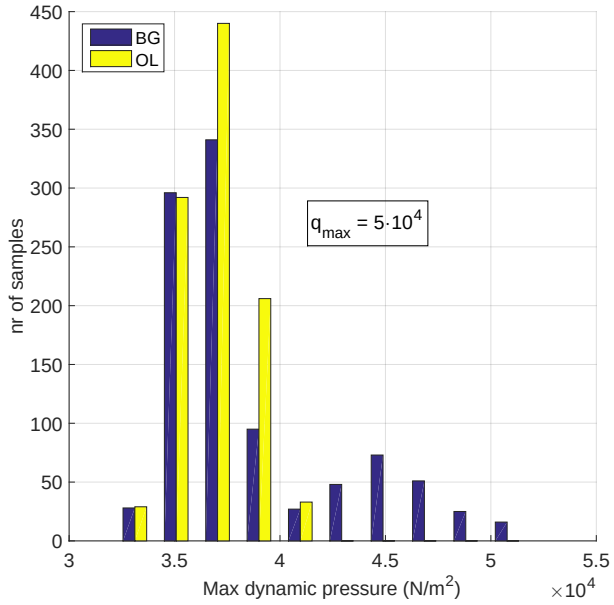


Figure 11.50: Monte-Carlo campaign IV - backup guidance vs open loop: dynamic pressure peaks.

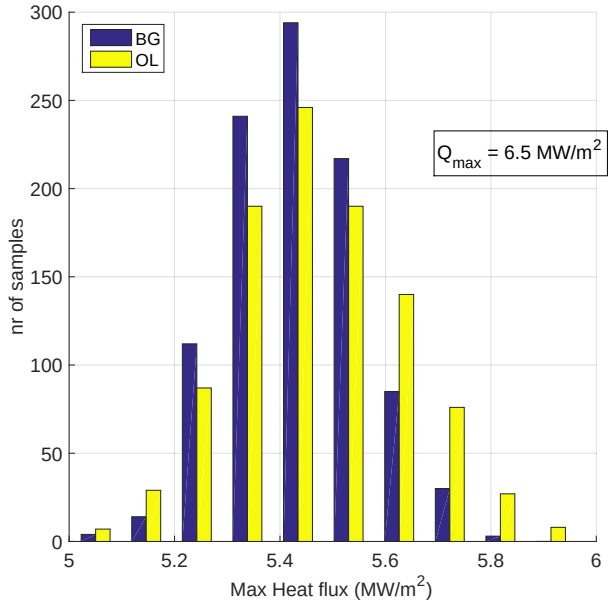


Figure 11.51: Monte-Carlo campaign IV - backup guidance vs open loop: heat flux peaks.

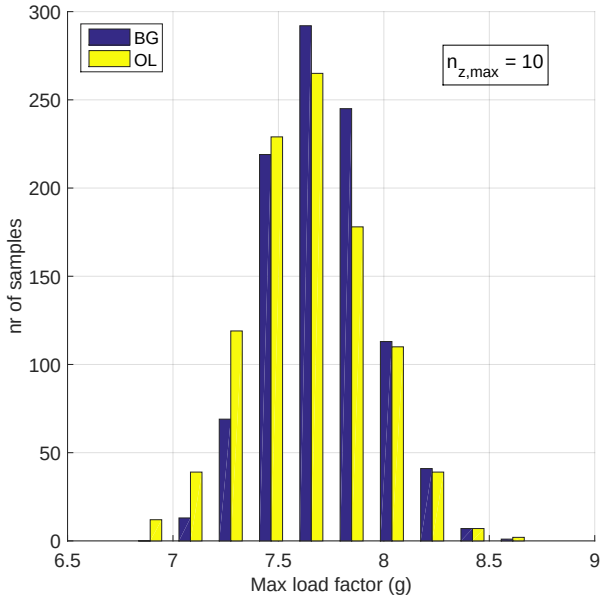


Figure 11.52: Monte-Carlo campaign IV - backup guidance vs open loop: vertical load factor peaks.

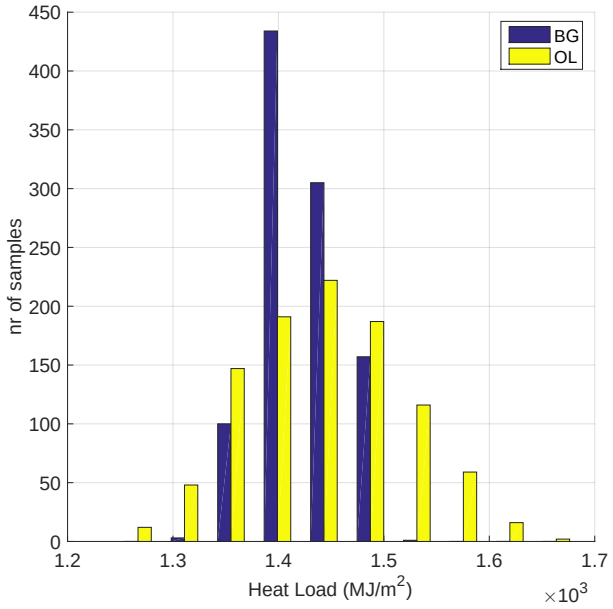


Figure 11.53: Monte-Carlo campaign IV - backup guidance vs open loop: heat load peaks.

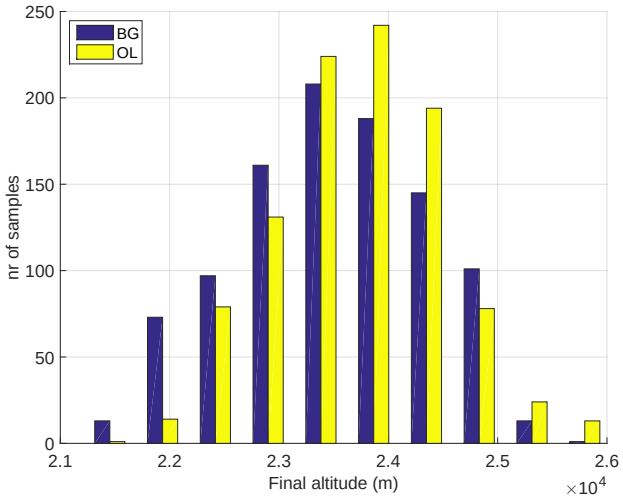


Figure 11.54: Monte-Carlo campaign IV - backup guidance vs open loop: final dispersions on altitude.

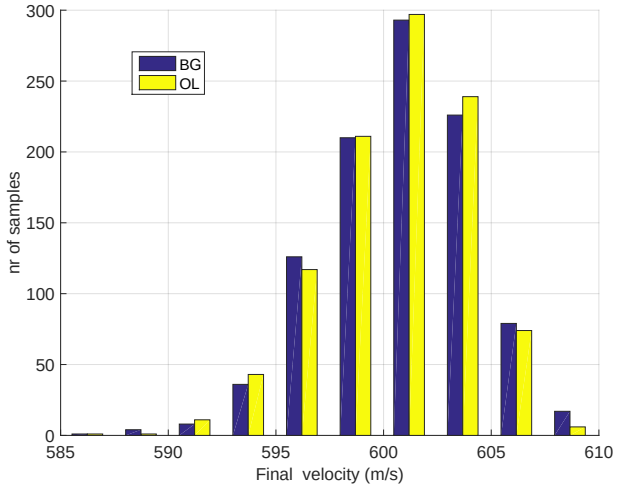


Figure 11.55: Monte-Carlo campaign IV - backup guidance vs open loop: final dispersions on velocity.

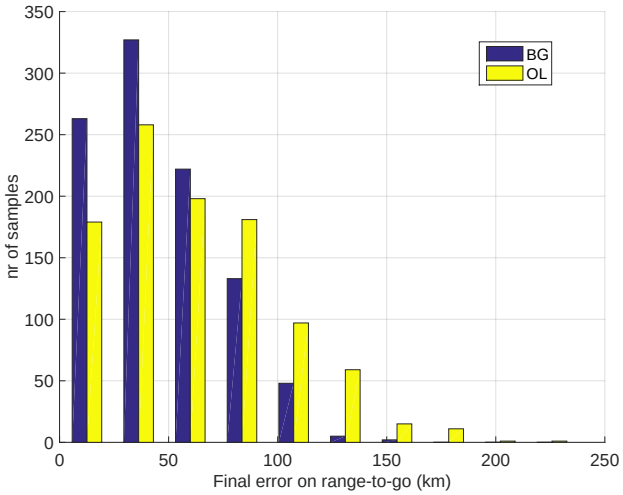


Figure 11.56: Monte-Carlo campaign IV - backup guidance vs open loop: final dispersions on range-to-go.

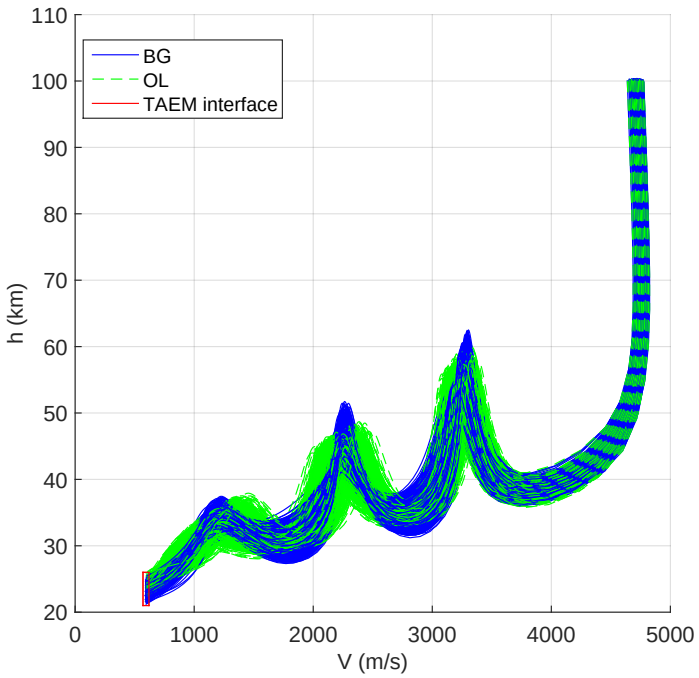


Figure 11.57: Monte-Carlo campaign IV - backup guidance vs open loop: h-V plane.

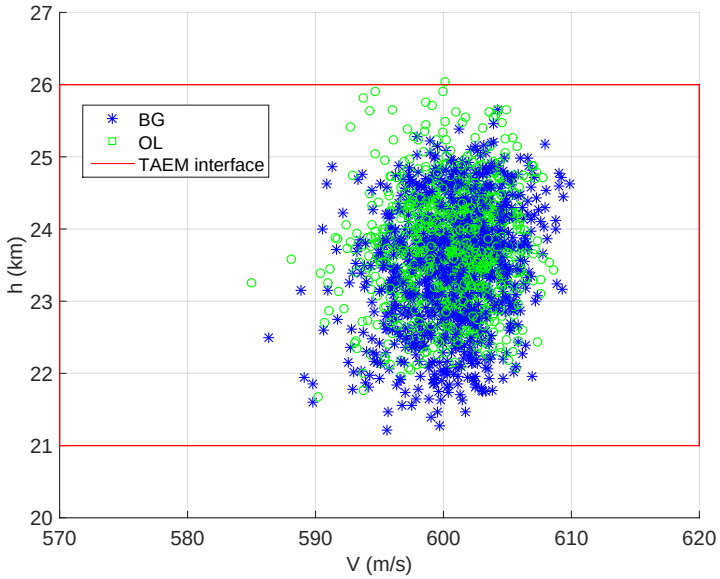


Figure 11.58: Monte-Carlo campaign IV - backup guidance vs open loop: h-V interface.

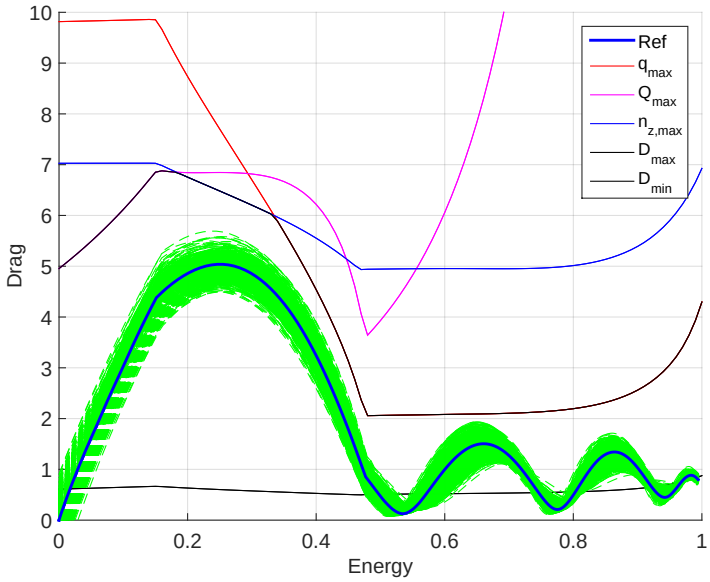


Figure 11.59: Monte-Carlo campaign IV - backup guidance vs open loop: drag vs energy envelope.

11.6 Comparison MG vs BG guidance

In this section a comparison between the main guidance and the backup guidance schemes is discussed. The analysis is done for the case associated with the use of the perturbed US76 profile. In this case, we will compare the two different guidance strategies to emphasize the behavior of the two systems.

11.6.1 Dispersion error

In terms of final dispersions, the main guidance behaves better than the backup guidance. While the overall dispersion area is practically the same ($\cong 22000$ km^2, Fig. 11.60, if we look at the range-to-go (Fig. 11.61), the main guidance ensures a larger number of cases having a shorter range-to-go w.r.t. the reference final longitude and latitude. This is very important, because for SHEFEX-3 and ReFEX, the recovery of the vehicle is one of the key-requirements, and a reduced range-to-go makes the recovery operations more efficient, and therefore cheaper. This is emphasized by Table 11.7, where we can observe a larger number of cases falling into the smallest circle (60.5% of cases for the main guidance against 55.8% for the backup guidance). Moreover, for the main guidance more than half (52.3%) of the case arrives at a distance of less than 42 km w.r.t. the nominal TAEM, against only 46% of the cases associated with the use of the backup guidance. If we limit the analysis to the cases falling within a distance of less than 30 km, the percentages are 31.1% for the main guidance against 25.6% for the backup guidance.

11.6.2 Constraints

For what regards the constraints, the two systems satisfy all the requirements. Figs. 11.62 through 11.65 show a comparison between main guidance and backup guidance in terms of maximum dynamic pressure, heat flux, vertical load factor, and heat load, respectively. In terms of dynamic pressure, the main guidance is more efficient. Indeed, from the analysis of Fig. 11.62 one can see that, even if both the systems stay within the limits, the backup guidance reaches larger peaks. This is due to a non-optimal angle of attack strategy, which uses a basic proportional law, and is not fully integrated within the overall backup guidance scheme, as is the case of the main guidance. For the heat-flux (Fig .11.63), and the load factor (Fig. 11.64), the distributions are more regular, with a slightly wider distribution for what regards the main guidance w.r.t. the backup guidance, around the same values, that is 5.5 MW/m^2 for the heat flux, and 7.85 g for the vertical load factor. The same conclusion can be drawn for the heat-load

Table 11.7: Dispersion Distribution - 1000 cases.

Mode / Radius	50×50 km^2	100×100 km^2	150×150 km^2	outside
MG	605	340	55	0
BG	558	392	50	0

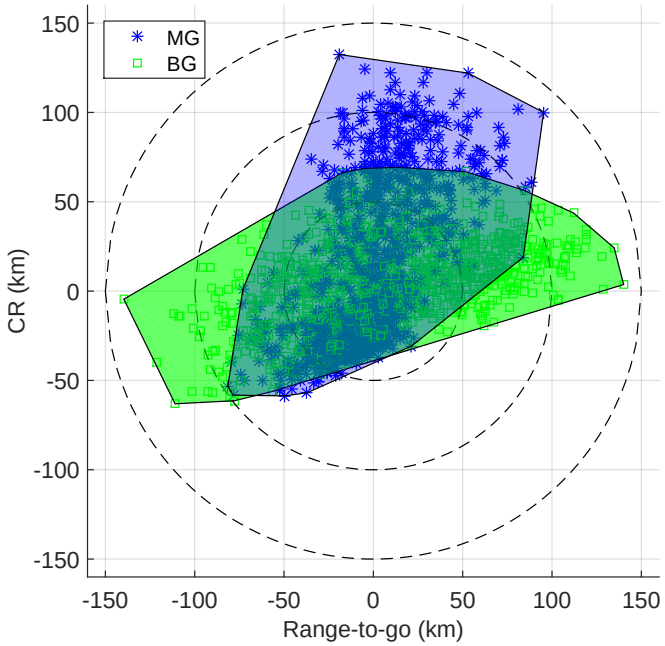


Figure 11.60: Performance comparison - main guidance vs backup guidance: dispersion area in DCA coordinates.

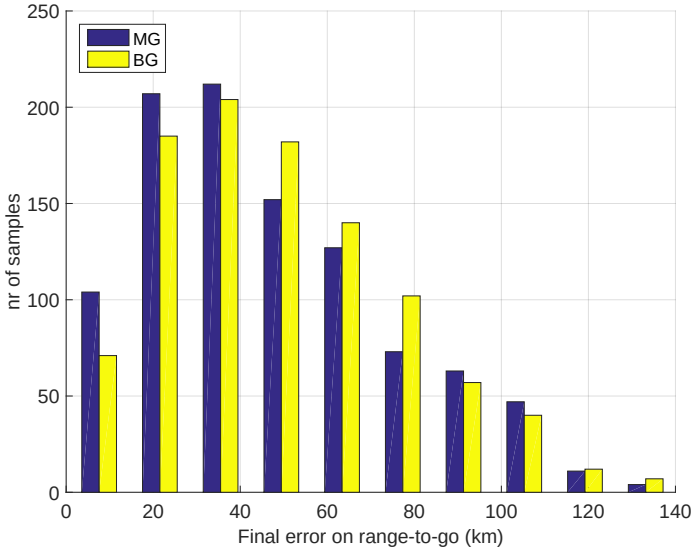


Figure 11.61: Performance comparison - main guidance vs backup guidance: final dispersions on range-to-go.

(Fig. 11.65). The peaks are between 1,400 and 1,450 MJ/m^2 for both the systems, with a slightly wider distribution of the values associated with the main guidance. In these plots we can see the different nature of the two guidance systems. While the main guidance is a state-tracking strategy, the backup guidance follows a constraint-tracking approach, which explains why the constraints peaks distributions are reduced.

Given the priority of the mission, which remains the reduction of the final dispersion while satisfying all the other requirements, the main guidance provides better results with respect to the backup guidance. It is also important to stress that, for such a constrained mission, without the "help" of SPARTAN, it would have been hard to design a feasible and satisfactory drag-energy profile. In other words, while the main guidance has been designed from scratch, the design of the backup guidance has been aided by the main guidance.

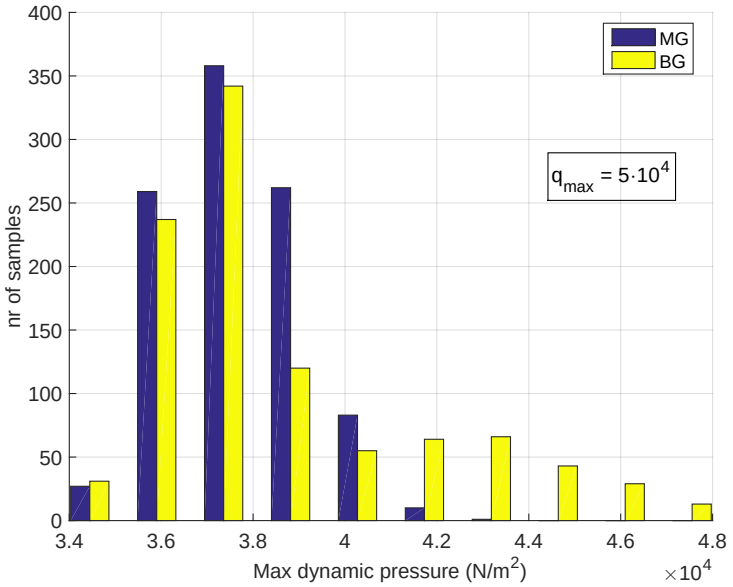


Figure 11.62: Performance comparison - main guidance vs backup guidance: dynamic pressure peaks.

11.6.3 Longitudinal Error

Finally, the two systems are equivalent from the point of view of the longitudinal error, as it can be seen from Figs. 11.66, and 11.67, in virtue of the Mach = 2 constraint. At the TAEM interface, maximum variations in terms of velocity are in the order of ± 5 m/s for the main guidance, and ± 3 m/s for the backup guidance, while for the altitude the distributions are almost the same, with the smaller values achieved by the backup guidance, and the larger values achieved by the main guidance. All of them are within the prescribed boundaries. These results can also be seen in Figs. 11.68 and

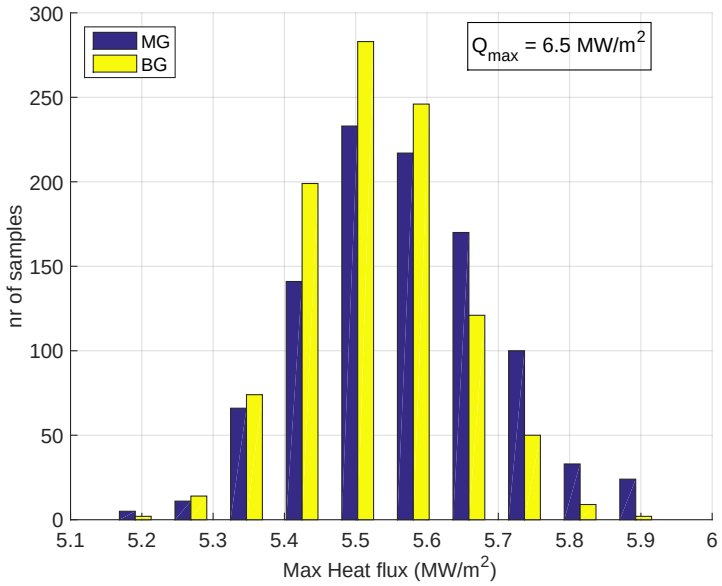


Figure 11.63: Performance comparison - main guidance vs backup guidance: heat flux peaks.

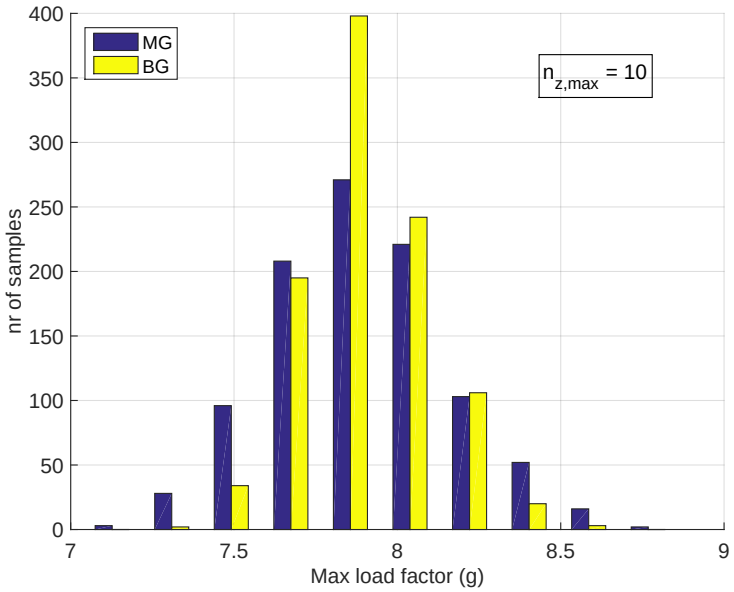


Figure 11.64: Performance comparison - main guidance vs backup guidance: vertical load factor peaks.

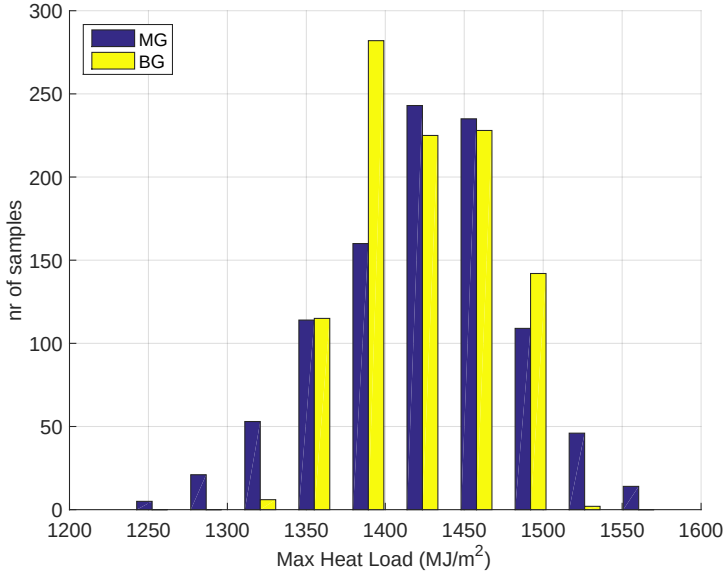


Figure 11.65: Performance comparison - main guidance vs backup guidance: heat load peaks.

11.69, where the trajectories are described on the altitude-velocity plane. An important conclusion, which can be drawn from this plot is that, even if numerically the backup guidance seems to be closer to the requirements, the main guidance provides much smoother trajectories. Indeed, in Fig. 11.68, the spikes during the flight associated with the backup guidance do not appear in the main guidance results.

To conclude, it is possible to state that both the systems provide valid results. However, given its higher flexibility, the main guidance provides better performance in terms of range-to-go. At the same time, the states and the trajectories are smoother than the ones obtained with the backup guidance. For brevity this analysis has been limited to the perturbed US76 model. However, from the results of the MC campaigns described in Sec. 11.3 and 11.5 it is possible to see the limits of the backup guidance applied to a scenario like the one treated in this work. The non-optimality of the angle-of-attack control law leads, in presence of strong uncertainties, to the risk of violations of the maximum allowed dynamic pressure. This does not happen in any case associated with the main guidance. Moreover, as already stated, while for other scenarios[2,37,41,52] it is possible to design the drag-energy profile analytically, this is not possible in this case. Therefore, the optimal drag-energy profile derived from the main guidance is used as basis to build the backup guidance scheme, which is in that sense not completely independent from the main guidance. Another strong point of the main guidance is that, if the perturbations become worse, the number of cases failing the requirements increases[1] but no drastic changes of the behavior of the system appear.

[1] As example, simulations with perturbations on the atmospheric density up to ±50% were performed,

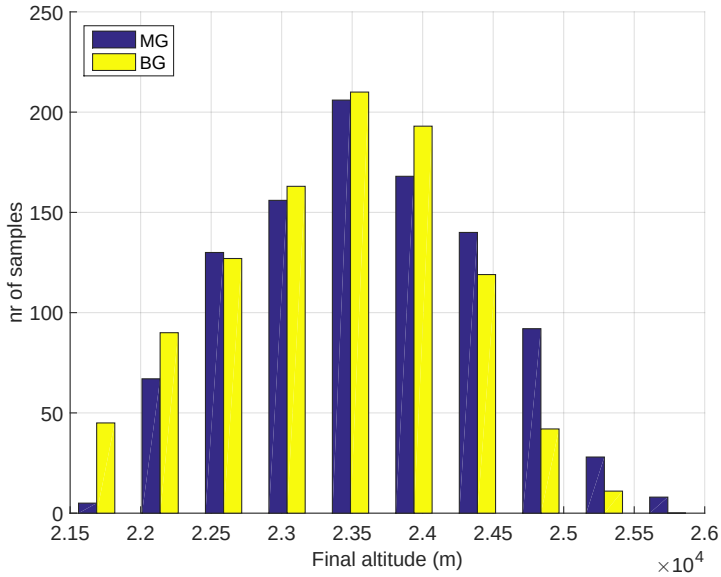


Figure 11.66: Performance comparison - main guidance vs backup guidance: final dispersions on altitude.

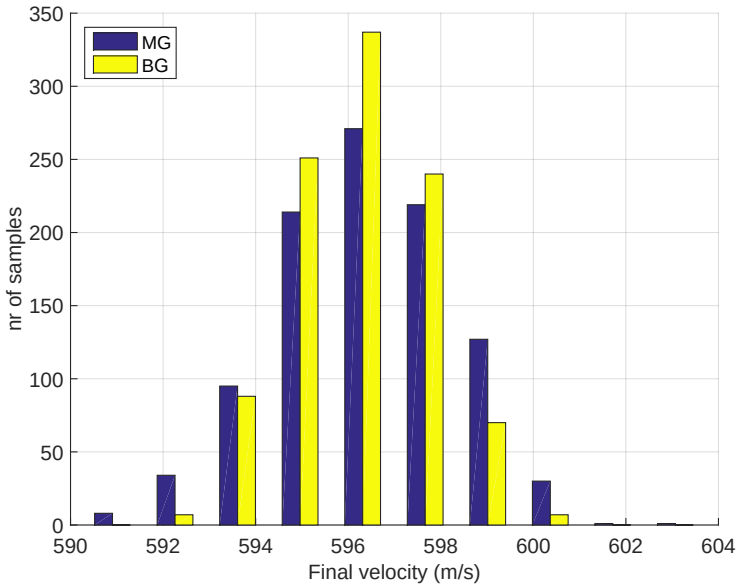


Figure 11.67: Performance comparison - main guidance vs backup guidance: final dispersions on velocity.

having as a result an increase up to 39.2% of the violations in terms of altitude at Mach = 2. However, the states are still smooth and consistent.

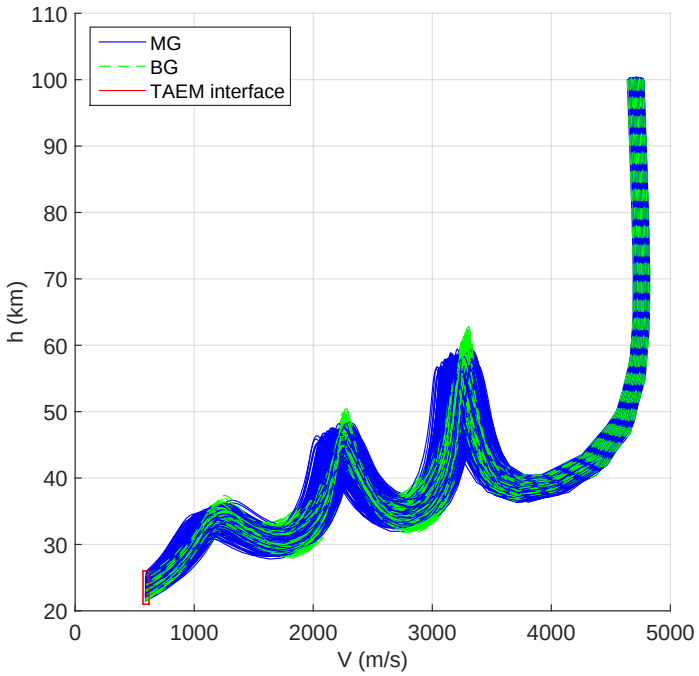


Figure 11.68: Performance comparison - main guidance vs backup guidance: h-V plane.

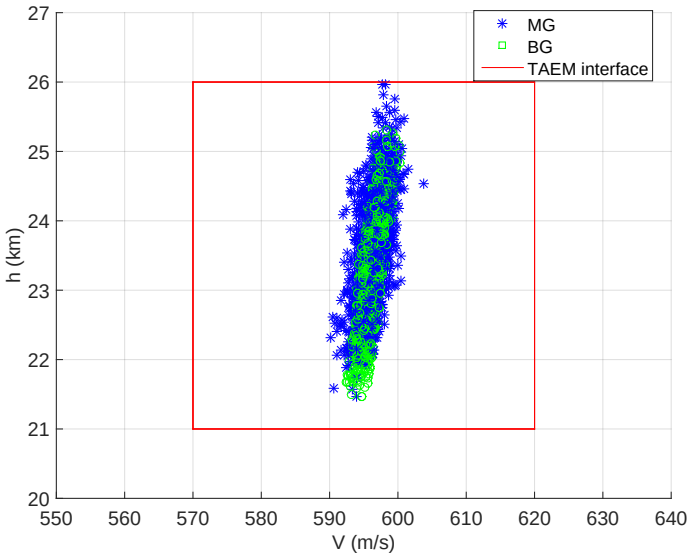


Figure 11.69: Performance comparison - main guidance vs backup guidance: h-V interface.

Chapter 12

Conclusions and Lesson Learned

12.1 Conclusions and Lesson Learned

12.1.1 Verification of the requirements

In this section the requirements defined in Table 2.4 are checked. This check is performed with the requirement compliance matrix reported in Table 12.1. All of the requirements are satisfied. Violations were observed only in two cases, both associated with the use of the perturbed atmospheric model NRLMSISE-00. The first violation regards the main guidance. Indeed, in one case the condition M = 2 was met at an altitude of about 170 m above the upper limit. Since a criterion of acceptance of 3σ was required, the requirement is considered fully satisfied. The second violation is associated with the backup guidance, and specifically, with the maximum value of the dynamic pressure, which shall not exceed $5 \cdot 10^4$ N/m^2.

In 16 cases, over a campaign of 1000 runs, this limit has been violated. This is due to the limits of the drag-energy scheme, which, for this scenario, is less able to deal with strong off-nominal conditions and for this reason, this specific requirement is considered only partially fulfilled. The main guidance satisfies this limits in all of the 1000 cases, instead. All the other requirements listed in Table 12.1 are fully satisfied.

Table 12.1: SHEFEX-3 Guidance Requirements.

Requirement	**Compliance**			**Comments**
	Full	Partial	None	
Req-G001		✓		MG fully satisfies the requirement. Because of the nature of the scenario, BG guidance fails in satisfying this requirement in 1.6% of the cases.
Req-G002	✓			Max value found equal to 5.9 MW/m^2.
Req-G003	✓			Max value found equal to 8.65 g.

Continued on next page

Table 12.1 -- continued from previous page

Requirement	Compliance			Comments
	Full	Partial	None	
Req-G004	✓			Included as terminal condition for each run.
Req-G005	✓			BG fully satisfies the requirement in all of the cases. MG does not satisfy this requirement only in one case over 1000 runs (error within 3σ).
Req-G006	✓			Min value found equal to 584.97 m/s and max value equal to 612.05 m/s.
Req-G007	✓			Max distance found w.r.t. the nominal target equal to 143 km.
Req-G008	✓			Included in the initial dispersions.
Req-G009	✓			Included in the initial dispersions.
Req-G010	✓			Included in the initial dispersions.
Req-G011	✓			Included in the initial dispersions.
Req-G012	✓			Included in the initial dispersions.
Req-G013	✓			Included in the initial dispersions.
Req-G014	✓			Included in the mass dispersions.
Req-G015	✓			Both the US-76 and NRLMSISE-00 atmospheric density have been perturbed up to ±20%. In case of NRLMSISE-00 variations in terms of F10.7 and magnetic indices have been considered too.
Req-G016	✓			Included in the aerodynamic dispersions.
Req-G017	✓			Included in the aerodynamic dispersions.
Req-G018	✓			These variations are in line with what is provided by the HWM07 model.
Req-G019	✓			Included in the controller design.
Req-G020	✓			Included in the controller design.
Req-G021	✓			Included in the controller design.
Req-G022	✓			Included in the controller design.
Req-G023	✓			Included in the controller design.
Req-G024	✓			Included in the controller design.
Req-G025	✓			Included in the controller design.
Req-G026	✓			Included in the controller design.
Req-G027	✓			Included in the controller design.
Req-G028	✓			1000 cases have been run for this MC campaign for each of the two guidance systems.
Req-G029	✓			1000 cases have been run for this MC campaign for each of the two guidance systems.

Continued on next page

Table 12.1 -- continued from previous page

Requirement	Compliance			Comments
	Full	Partial	None	
Req-G030	✓			Included in the gravity model.
Req-G031	✓			Included in the gravity model.
Req-G032	✓			Not included in the controller design.
Req-G033	✓			Included in the Monte-Carlo campaigns.
Req-G034	✓			Included in the controller design.
Req-G035	✓			Included in the Monte-Carlo campaigns
Req-G036	✓			The two guidance systems are completely independent.

12.1.2 Conclusions

This thesis describes in detail two different guidance systems developed for SHEFEX-3, which can potentially be used for the SHEFEX program evolution, ReFEX. Realistic uncertainties and limitations associated with a real mission have been taken into consideration. Many of the simplifications made in literature have been rejected, making most of the work potentially reusable for other real missions and scenarios. In particular, the main guidance can be seen as an on-board trajectory generator, which can handle hard-constrained scenarios. For instance, in this case the classical bank-reversal strategy to limit the lateral error, is not sufficient, and a different strategy was needed.

The main guidance was able to improve the performance and to reduce the lack of performance, which would occur in case only classical strategies had been used. The result is an adaptive, real-time capable algorithm, able to deal with strong initial dispersions. However, these dispersions were not the only uncertainties acting on the system, as we have seen in Chapter 11. For this reason, the on-board adaptive trajectory generation algorithm has been coupled to a robust, nonlinear controller, based on the use of sliding mode control, to reduce the errors in the limits of the vehicle's capabilities.

To assess the behavior of the main guidance, and, at the same time, to increase the redundancy of the guidance system, a second, completely independent, guidance system has been designed. Even though this system is based on the same ideas which led to the Space Shuttle entry guidance, and its evolutions, the specific characteristics of the SHEFEX-3 mission, together with the requirements for what regards the effort to model the atmosphere and the vehicle (e.g., the aerodynamic database), required a redesign of the classical drag-energy scheme.

As it has been reported in Chapter 10, most of the simplifications associated with these schemes have been here rejected, to have a more general scheme, which reduces to one of the well-known drag-energy schemes where the same assumptions are made. However the scheme has general validity, and can therefore be used for other missions, vehicles, and even celestial bodies having an atmosphere, such as Mars.

The two different architectures have been compared in terms of longitudinal performance and final dispersions. Specifically, separated Monte-Carlo campaigns associated with the use of perturbed US-76 and NRLMSISE-00, as well as many other uncertainties and perturbations, have been performed. A specific effort was made to code realistic uncertainties, such as time-varying density profiles, time-varying lift and drag coefficients, and accurate models such as the HWM07 for representing the wind.

The algorithms have been tested in conditions where a strong asymmetry due to the initial heading error, wide dispersions at EIP were considered, and the results are satisfactory according to the scenario limits, and the vehicle capabilities. Indeed, especially when the standard atmosphere US-76 has been used, both the developed algorithms satisfy all the requirements defined in Table 2.4.

In terms of performance, the main guidance shows better results w.r.t. the backup guidance, especially in terms of range-to-go, as a larger number of cases fall within the smallest circle defined to classify the final dispersion of the vehicle. Moreover, the integration of the angle of attack in the guidance design leads to smoother trajectories, and no violations of the constraints have been met.

To answer the question made at the beginning of the work, that is:

How can optimal control theory be integrated into an entry guidance scheme, which is real-time capable, and able to deal with a highly-constrained scenario? If successful, is this method able to compete with the current state of the art for the entry guidance algorithms (i.e., the drag-tracking methods)?

it is now possible to state that the optimal control-related techniques are mature enough to be considered as a valid alternative to the drag-tracking methods, and the results show that, for the current scenario, they behave better than the classical methods, especially in terms of final dispersion, which is a key-requirement for SHEFEX-3. This is actually one of the most appealing features of the proposed main guidance.

The other strong advantage of using it comes from its real-time capability. While real-time optimization will be hopefully possible with the next generation of on-board computers, the method developed here makes possible the use of offline optimal trajectories, which are opportunely combined, taking the limitations of the current on-board architectures into account. Indeed, none of the parts of the developed guidance system requires iterative procedures, but they are based on simple algebraic operations, both for the adaptive feedforward and the robust feedback guidance subsystems. Therefore, it is always possible to predict the number of operations required for each cycle of computation, and the proposed algorithms are suitable for real-time applications. CPU time analyses performed during the development of the work have shown that an entire trajectory can be computed with a normal laptop in about 130 milliseconds, while the computation of a corresponding optimal control solution may require a CPU time between 60 and 600 times larger.

Another advantage associated with the use of the main guidance is the use of the AMPI on one side, and the SMC on the other side. Specifically, we have already

illustrated in Chapter 8 the advantages in terms of CPU burden and CPU memory associated with the use of the AMPI algorithm. Another advantage comes from the SMC, which is a nonlinear control strategy, and does not require any gain-scheduling technique. Therefore, the on-board memory requirements are further reduced, making the algorithm appealing not only in terms of CPU burden, but also in terms of required data storage.

Monte-Carlo campaigns associated with the atmospheric model NRLMSISE-00 show that in case of different perturbations, while the main guidance still satisfies all the requirements except for one unique case (success rate 99.9%), the backup guidance has a lower success rate (98.4%), violating the 3σ criterion. The reason for this failure resides mainly in the limits of the scenario and the vehicle. For instance, a better characterization of the atmospheric models with atmospheric balloons (e.g., as it has been done with the IXV experiment) may significantly reduce the atmospheric uncertainties, leading consequently a strong improvements in the performance. This effect is reduced in the cases associated with the main guidance for the intrinsic robustness of the SMC. However, corresponding improvements in the results can be obtained with the backup guidance method as well in case a stronger effort for characterizing the models is performed, or alternatively, if better conditions for the flight-path angle are achieved at the entry interface. Indeed, this leads to reduced oscillations of flight-path angle, altitude, and drag profile, having as a result a significant reduction of the complexity of the drag-energy profile design, which may be made adaptive, with a consequent improvement of the performance.

It was not certain at the beginning of the work whether both the adaptive feedforward guidance and the robust feedback guidance based on the use of sliding mode control may provide good results for the entry guidance. Indeed, the multivariate interpolation is a completely new way to approach the on-board trajectory generation process, and its combination with pseudospectral methods has brought properties never explored previously. The use of sliding mode control itself has not been yet deeply exploited in the frame of the entry guidance, and the work started here can be used as basis for further studies and improvements in this direction.

Finally, SPARTAN is fully functional, and its extension and consolidation is at the moment one of the objectives of the author. While the exact differentiation capability (based on Dual-number theory) has already been successfully implemented and validated, the tool is at moment not yet able to work with multiphase problems (a typical example is the ascent trajectory of a multi-stage rocket). However, the technological maturity that has achieved in the last three years makes it an important outcome of this work.

12.1.3 Lesson Learned

While the crossrange capabilities have been here widely demonstrated, it is worth to say that the choice of this scenario has a clear impact on all the other performances of the system. In other words, it would be possible to reduce the error on longitudinal performance and final dispersions if the required nominal crossrange is reduced. This is indeed what is usually done with other autonomous vehicles such as the Mars Science

Laboratory, where the initial heading error is already in the allowed corridor.

If stricter requirements on the dispersions are required, another relevant aspect, which can be improved, is the characterization of the uncertainties: measurements of the atmospheric density some days (or hours) before the flight, and more experimental data to characterize the aerodynamic database of the vehicle can help to reduce the uncertainties, and have a direct impact on the performance of the guidance system. This approach has been successfully followed in the past with the Space Shuttle flights, and with the recent IXV, where the last measurements of the atmosphere were taken 4 hours before the flight.

Finally, it is worth to say that the backup guidance does not seem to fit very well with an oscillatory entry like the one studied in this work. Indeed, changes in the flight-path angle induced by this type of entry, and the consequent drag dynamics, make hard, especially in presence of wide disturbances as in this case, to bring the drag-dynamics error asymptotically to zero. This aspect can be improved by having a better entry interface, especially in terms of flight-path angle, (almost 0 deg, as in the case of the aforementioned IXV), or, in case of negative flight-path angle, higher control capability in terms of bank angle (as in the case of X-33). In both cases indeed, the drag dynamics would be simpler, and the performance would improve consequently. Alternatively, as already mentioned, greater efforts may be made to reduce the uncertainties coming from the atmosphere and the aerodynamics of the vehicle.

References

[1] Bogner, I., ''Description of Apollo Entry Guidance,'' Tech. rep., NASA, 1963. 14, 74, 75, 194

[2] Harpold, J. D. and Graves, C. A., ''Shuttle Entry Guidance,'' *Journal of Astronautical Sciences, Vol.27 No.3, May-June 1979, pp.239-268*, 1979. 14, 66, 74, 75, 76, 77, 78, 79, 82, 187, 190, 194, 199, 203, 255

[3] Mendeck, G. and Craig, L., ''Entry Guidance for the 2011 Mars Science Laboratory,'' *AIAA Atmospheric Flight Mechanics Conference*, 2011. 14

[4] Martin, M. S., Mendeck, G., Brugarolas, P. B., Singh, G., and Serricchio, F., ''Reconstructed Flight Performance of the Mars Science Laboratory Guidance, Navigation, and Control System for Entry, Descent, and Landing,'' *Advances in the Astronautical Sciences Guidance, Navigation and Control*, 2014. 14

[5] Bedrossianand, N., Bhatt, S., Lammers, M., Nguyen, L., and Zhang, Y., ''First Ever Flight Demonstration of Zero Propellant Maneuver Attitude Control Concept,'' *AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, USA*, 2007. 16, 90, 111

[6] Elissar, ''Description of DIDO Optimal Control software,'' June 2015. 16, 91

[7] Mease, K. D., Chen, D. T., Teufel, P., and Schöneberger, H., ''Reduced-Order Entry Trajectory Planning for Acceleration Guidance,'' *Journal of Guidance, Control, and Dynamics, Vol.25 No.2, March-April*, 2002. 17, 29, 64, 80, 81, 82, 194, 197, 199, 200, 203, 206

[8] Lu, P. and Hanson, J. M., ''Entry Guidance for the X-33 Vehicle,'' *Journal of Spacecraft and Rockets, Vol.35 No.3, May-June*, 1998. 17, 29, 64, 83

[9] Lu, P., ''Entry Guidance: A Unified Method,'' *Journal of Guidance, Control, and Dynamics, Vol.37 No.3, May-June*, 2014. 17, 85, 86, 87

[10] D'Onofrio, V., Sagliano, M., and Arslantas, Y., ''Exact Hybrid Jacobian Computation for Optimal Trajectory Computation via Dual Number Theory,'' *AIAA Science and Technology Forum and Exposition*, 2016. 18, 51

[11] Huneker, L., Sagliano, M., and Arslantas, Y., ''SPARTAN: An Improved Global Pseudospectral Algorithm for High-Fidelity Entry-Descent-Landing Guidance Analysis,'' *30th International Symposium on Space Technology and Science, Kobe, Japan, 2015*, 2015. 18, 112

[12] Arslantas, Y. E., Oehlschlägel, T., Sagliano, M., Theil, S., and Braxmaier, C., ''Safe Landing Area Determination for a Moon Lander by Reachability Analysis,'' *International Astronautical Conference. IAC-14-C.1.7.2, Toronto, Canada, 2014*, 2014. 18, 108

[13] Arslantas, Y. E., Oehlschlägel, T., Sagliano, M., Theil, S., and Braxmaier, C., ''Approximation of Attainable Landing Area of a Moon Lander by Reachability Analysis,'' *17th International Conference and Control (HSSC), Berlin, Germany*, 2014. 18, 108, 112

[14] Sagliano, M. and Theil, S., ''Hybrid Jacobian Computation for Fast Optimal Trajectories Generation,'' *AIAA Guidance, Navigation, and Control Conference, Boston, USA,*, 2013. 18, 51, 111

[15] Sagliano, M., ''Performance analysis of linear and nonlinear techniques for automatic scaling of discretized control problems,'' *Operations Research Letters, Vol.42 Issue 3, May 2014, pp. 213-216*, 2014. 18, 111

[16] Sagliano, M., Samaan, M., Theil, S., and Mooij, E., ''SHEFEX-3 Optimal Feedback Entry Guidance,'' *AIAA SPACE 2014 Conference and Exhibition, San Diego, USA*, 2014. 18, 29, 92, 166

[17] Sagliano, M., Oehlschlägel, T., Theil, S., and Mooij, E., ''Real Time Adaptive Feedforward Guidance for Entry Vehicles,'' *3rd CEAS Eurognc conference, Toulouse*, 2015. 18, 30, 162, 166

[18] Sagliano, M., Mooij, E., and Theil, S., ''Onboard Trajectory Generation for Entry Vehicle via Adaptive Multivariate Pseudospectral Interpolation,'' *AIAA Science and Technology Forum and Exposition*, 2016. 18

[19] Weihs, H., ''The SHEFEX Story: A Historical Review 2001-2014,'' *5th ARA Days*, 2015. 23, 25

[20] Eggers, T., Braunschweig, D., BRAUnschweig, D., BRAUnschweig2, D., BRAUnschweig3, D., and BRAUnschweig4, D., ''The SHEFEX Flight Experiment: Pathfinder Experiment for a Sky Based Test Facility,'' *14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference*, No. AIAA 2006-7921, AIAA, 2006. 23

[21] DLR, ''Flugexperiment SHEFEX erfolgreich gestartet,'' October 2005. 23, 24

[22] Bollino, K. P., Ross, I. M., and Doman, D., "Optimal Nonlinear Feedback Guidance for Reentry Vehicles," *AIAA Guidance Navigation, and Control Conference and Exhibit, Keystone, USA*, 2006. 27, 29, 91, 92, 93

[23] ESA, "Margin Philosophy for Science Assessment Studies," 2012. 30, 211

[24] Drob, D. P., Emmert, J., Crowley, G., Picone, J. M., Shepherd, G. G., Skinner, W., and Hays, P., "An empirical model of the Earth's horizontal wind fields: HWM07," *Journal of Geophysical Research, Vol. 113, No.A12304, 2008*. 31, 40, 42, 43, 57, 212

[25] Picone, J. M., Hedin, A. E., Drob, D. P., and Aikin, A. C., "NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues," *Journal of Geophysical Research, Vol. 107, No.A12,*, 2002. 31, 38

[26] Neeb, D. and Gülhan, A., "Experimentelle Bestimmung der aerodynamischen Beiwerte von SHEFEX II im H2K," Tech. Rep. DLR Technical Report 32418 2009A36, DLR, 2012. 31, 43, 212

[27] Gallais, P., *Atmospheric Re-entry Vehicle Mechanics*, Springer, 2007. 35

[28] Mooij, E., "Adaptive Heat-Flux Tracking for Re-entry Guidance," *AIAA/AAS Astrodynamics Specialist Conference, San Diego, USA*, 2014. 35, 87, 89, 90

[29] Singh, B. and Bhattacharya, R., "Optimal Guidance of Hypersonic Vehicles Using B-Splines and Galerkin Projection," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008. 35

[30] NASA, "U.S. Standard Atmosphere 1976," Tech. Rep. NASA Technical Memorandum X-74335, 1976. 36, 94

[31] Hedin, A. E., "MSIS-86 thermospheric model," *Journal of Geophysical Research, Vol. 92, May 1,*, 1987. 37

[32] Department of Defense, "World Geodetic System 1984," Tech. Rep. NIMA TR8350.2 3rd Ed., 1984. 40

[33] Fike, J. and J.Alonso, "The Development of Hyper-Dual Numbes for Exact Second-Derivative Calculations," *49th AIAA Aerospace Sciences meeting including the New Horizons Forum and Aerospace Exposition , Orlando, USA, 2011*, No. AIAA paper 2011-886. 51, 137

[34] D'Onofrio, V., *Implementation of Advanced Differentiation Methods for Optimal Trajectory Computation*, Master's thesis, University of Naples Federico II, Naples, 2015. 51, 137

[35] Broquet, R., Perrimon, N., Polle, B., Hyounet, P., Krauss, P., Drai, R., Voirin, T., and Fernandez, V., "HiNAV Inertial / GNSS Hybrid Navigation System for

launchers and re-entry vehicles,'' *Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010 5th ESA Workshop on*, dec. 2010, pp. 1 --6. 55

[36] Mooij, E., *The Motion of a Vehicle in a Planetary Atmosphere*, Vol. Astrodynamics and Satellite Systems 01 of *08*, Delft University Press, Delft, The Netherlands, Mekelweg 4, 2628, CD, Delft, The Netherlands, 1997. 60, 67, 70, 72

[37] Saraf, A., Leavitt, J. A., Chen, D. T., and Mease, K. D., ''Desing and Evaluation of an Acceleration Guidance Algorithm for Entry,'' *Journal of Spacecraft and Rockets, Vol.41 No.6, November-December*, 2004. 66, 80, 81, 84, 187, 194, 197, 199, 255

[38] Roenneke, A. J. and Markl, A., ''Reentry Control to a Drag vs. Energy Profile,'' *AIAA Guidance, Navigation, and Control Conference, Monterey, USA*, 1993. 80, 84, 85, 197, 200

[39] Mease, K. D., Teufel, P., Schöneberger, H., Chen, D. T., and Bharadwaj, S., ''Reentry Trajectory Planning for a Reausable Launch Vehicle,'' *Atmospheric Flight Mechanics Conference, 1999*, 1999. 80, 81, 82, 83

[40] Lu, P., ''Entry Guidance and Control for Reusable Launch Vehicles,'' *AIAA Guidance, Navigation, and Control Conference, San Diego, USA*, 1996. 83, 85

[41] Lu, P., ''Entry Guidance and Trajectory Control for Reusable Launch Vehicle,'' *Journal of Guidance, Control, and Dynamics, Vol.20 No.1, January-February*, 1997. 83, 84, 255

[42] Roenneke, A. J. and Well, K. H., ''Nonlinear Drag-Tracking Control Applied to Optimal Low-lift Reentry Guidance,'' *AIAA Guidance, Navigation, and Control Conference, San Diego, USA,*, 1996. 84, 200, 203

[43] Lu, P., ''Predictor-Corrector Entry Guidance for Low lifting Vehicles,'' *Journal of Guidance, Control, and Dynamics, Vol.31 No.4, July-August*, 2008. 85

[44] Lu, P., ''Gliding Guidance of High L/D Hypersonic Vehicles,'' *AIAA Guidance, Navigation, and Control Conference, Boston, USA,*, 2013. 85, 86

[45] Mooij, E., ''Robustness Analysis of an Adaptive Re-entry Guidance System,'' *AIAA Guidance, Navigation, and Control Conference, San Francisco, USA*, 2005. 89, 90

[46] Fahroo, F., ''Advances in Pseudospectral Methods for Optimal Control,'' *AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, USA, 2008*. 90, 111

[47] Ross, I. M., Sekhavat, P., Fleming, A., and Gong, Q., ''Pseudospectral Feedback Control: Foundations, Examples and Experimental Results,'' *AIAA Guidance, Navigation, and Control Conference, Keystone, USA,*, 2006. 90, 111, 168

[48] Recasens, J. J., Chu, Q. P., and Mulder, J. A., ''Robust Model Predictive Control of a Feedback Linearized System for a Lifting-body Re-entry Vehicle,'' *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California*, 2005. 92

[49] Lu, W. M. and Bayard, D. S., ''Guidance and Control for Mars Atmospheric Entry: Adaptivity and Robustness,'' Tech. rep., Jet Propulsion Laboratory, California Institute of Technology, 1999. 92

[50] Shue, S., Sawan, M., and Rokhsaz, K., ''Mixed H2/H-infinity gain scheduling control of aircraft,'' *Guidance, Navigation, and Control Conference*, 1996. 92

[51] Shen, Z. and Lu, P., ''Onboard Generation of Three-Dimensional Constrained Entry Trajectories,'' *Journal of Guidance, Control, and Dynamics, Vol.26 No.1, January-February*, 2003. 92

[52] Mease, K. D., Chen, D. T., Tandon, S., Young, D. H., and Kim, S., ''A Three-Dimensional Predictive Entry Guidance Approach,'' *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000. 92, 199, 255

[53] Shtessel, Y. B., ''Nonlinear output tracking in conventional and dynamic sliding manifolds,'' *IEEE Transactions in Automatic Control, Vol.42 Issue 9, pp.1282-1286*, 1997. 92

[54] Shtessel, Y. B., Edwards, C., Fridman, L., and Levant, A., *Sliding Mode Control and Observation*, Birkhäuser, 2013. 92, 178, 184

[55] Bollino, K. P., *High-Fidelity Real-Time Trajectory Optimization for Reusable Launch Vehicles*, Ph.D. thesis, Naval Postgraduate School, Monterey, 2006. 93

[56] MathWorks Inc., ''Implement 1984 World Geodetic System (WGS84) representation of Earth's gravity,'' June 2015. 95

[57] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., *The Mathematical Theory of Optimal Processes*, 1962. 111

[58] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, 2^{nd} ed.*, SIAM, Philadelphia, 2010. 111, 112, 124, 127, 132, 145, 146, 147

[59] Garg, D., *Advances in Global Pseudospectral Methods for Optimal Control*, Ph.D. thesis, University of Florida, Gainesville, 2011. 111, 115, 150, 168

[60] Gong, Q., Ross, I. M., Kang, W., and Fahroo, F., ''Connections Between The Covector Mapping Theorem and Convergence of Pseudospectral Methods for Optimal Control,'' *Comput Optim Appl, 2008*, 2008. 111, 168

[61] Gill, P. E., Murray, W., and Saunders, M. A., *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, University of California, San Diego, USA, 2008. 112

[62] Wächter, A. and Biegler, L. T., ''On the implementation of an interior-point filter linesearch algorithm for large-scale nonlinear programming,'' *Math. Program. 106(1) , Springer-Verlag, New York, 2006*, 2006. 112

[63] Abramovitz, M. and Stegun, I. A., *Handbook of Mathematical Functions*, Dover Publications, 1695. 115, 120

[64] Martins, J. R. R., Sturdza, P., and Alonso, J. J., ''The Complex-Step Derivative Approximation,'' *ACM Transactions on Mathematical Software, Vol. 29, No. 3, September 2003, Pages 245262*, 2003. 120, 136, 137

[65] Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer-Verlag, New York, 1999. 124

[66] Rao, A. V., ''A Survey of Numerical Methods for Optimal Control,'' *AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 09-334, Pittsburgh, PA, August 10 - 13*, 2009. 124

[67] Herman, A. L. and Conway, B. A., ''Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules,'' *Journal of Guidance, Control, and Dynamics, Vol.19 No.3, May-June 1996*, 1996. 141, 142, 143

[68] DeBoor, C., *A Practical Guide to Splines*, Springer, New York. 167

[69] Fernandez, B. and Hedrick, K. J., ''Control of Multivariable Non-Linear Systems by the Sliding Mode Method,'' *Int. J. Control, Vol. 46, No. 3, pp-1019-1040*, 1987. 178, 183

[70] Hedrick, J. K. and Yip, P. P., ''Multiple Sliding Surface Control: Theory and Application,'' *Transactions of the ASME, Vol. 122, December 2000*, 2000. 178

[71] Filippov, A. F., *Differential Equations with Discontinuous Righthand Sides*, 1988. 178

[72] Perruquetti, W., *Sliding Mode Control in Engineering*, Marcel Dekker Inc, 2002. 178, 184

[73] Khalil, H. K., *Nonlinear System 3^{rd} Ed.*, Prentice Hall, New Jersey, 2002. 178

[74] Kol, I., Michor, P., and Slovak, J., *Natural operations in differential geomentry*, Springer-Verlag, 1993. 179

[75] Xu, H., Mirmirani, M. D., and Ioannou, P. A., ''Adaptive Sliding Mode Control Design for a Hypersonic Flight Vehicle,'' *Journal of Guidance, Control, and Dynamics, Vol.27 No.5, September-October 2004*, 2004. 180, 182

[76] Harl, N. and Balakrishnan, S., ''Reentry Terminal Guidance Through Sliding Mode Control,'' *Journal of Guidance, Control, and Dynamics, Vol.33 No.1, January-February 2010*, 2010. 180

[77] Furfaro, R. and Wibben, D. R., ''Mars Atmospheric Entry Guidance via Multiple Sliding Surface Guidance for Reference Trajectory Tracking,'' *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Minneapolis, USA, 2012*. 180

[78] Mooij, E., *Aerospace-Plane Flight Dynamics*, Ph.D. thesis, Delft University of Technology, Delft, 1998. 186

[79] Mooij, E., ''Passivity Analysis for Non-Linear, Non-Stationary Entry Capsules: Translational Motion,'' *Invited paper 2161, pp. 6549-6554, 18 th IFAC World Congress, Milano (Italy), August 28 - September 2*, 2011. 186

[80] Mooij, E., ''Characteristic Motion of Re-entry Vehicles,'' *AIAA Atmospheric Flight Mechanics Conference, Boston*, 2013. 186

[81] Tu, K. Y., Munir, M. S., and Mease, K. D., ''Drag-Based Predictive Tracking Guidance for Mars Precision Landing,'' *Journal of Guidance, Control and Dynamics, Vol.23 No. 4, July-August 2000, pp.620-628*, 2000. 187, 194, 203, 233

[82] Mease, K. D. and Kremer, J. P., ''Shuttle Entry Guidance Revisited Using Nonlinear Geometric Methods,'' *Journal of Guidance, Control, and Dynamics, Vol.17 No.6, November-December*, 1994. 194

[83] Roenneke, A. J., ''Adaptive On-board Guidance for Entry Vehicles,'' *AIAA Guidance, Navigation, and Control Conference, Montreal, Canada*, 2001. 203

[84] Bharadwaj, S., Rao, A. V., and Mease, K. D., ''Entry Trajectory Tracking Law via Feedback Linearization,'' *Journal of Guidance, Control, and Dynamics, Vol.21 No.5, September-October 1998*, 1998. 203

[85] ''Selected Geomagnetic and Solar Activity Indices,'' July 2015. 212

[86] ''Geomagnetic Data Indices Kp / Ap,'' July 2015. 212

[87] Benito, J. and Mease, K. D., ''Nonlinear Predictive Controller for Drag Tracking in Entry Guidance,'' *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, USA*, 2008. 233